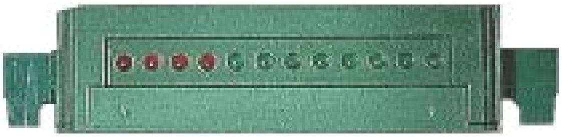
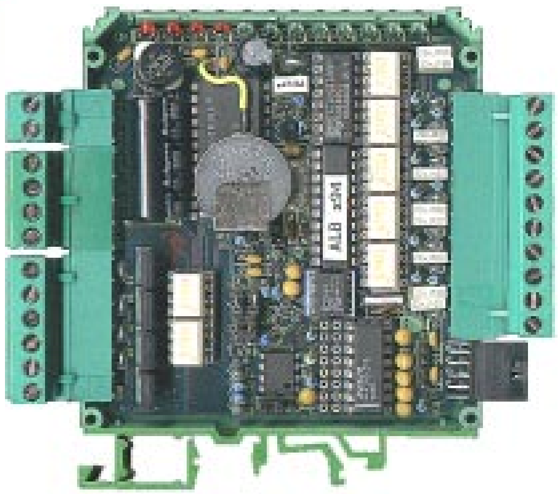
GPC® R/T94

General Purpose Controller
Relays or Transistors; 9 Inputs, 4 Outputs

MANUALE TECNICO







grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

GPC® R/T94 Edizione 3.40 *Rel. 25 Luglio 2001*

GPC® R/T94

General Purpose Controller
Relays or Transistors; 9 Inputs, 4 Outputs

MANUALE TECNICO

Modulo Intelligente **ABACO®** Block della Serie M; Ingombro di 22,5x82x90 mm; Contenitore per guide ad Ω tipo DIN 46277-1 e DIN 46277-3; CPU 89C2051 o 89C4051 a 14MHz; 2K o 4K Bytes di FLASH e 128 Bytes di SRAM; Codice compatibile con i μP della famiglia 51; Real Time Clock con 256 Bytes di SRAM interna; Circuiteria di Back-Up per RTC e SRAM, tramite batteria al LITIO; E^2 seriale da 1K Bytes; 9 linee di ingresso NPN, galvanicamente isolate; Linea di INT esterno, galvanicamente isolata; 4 linee di uscita: A Relay da 5A nella versione R, Optoisolote, di tipo NPN con Darlington da 4A, 45 Vdc privi di radiatore e con diodo di ricircolo, nella versione T; LEDs, di visualizzazione dello stato di tutte le linee di I/O; Collegamento delle I/O tramite connettori a rapida estrazione; 2 linee di I/O TTL; Registri di Timer-Counter da 16 bits; Circuiteria per A/D converter con risoluzione di 11 bits, tempo di conversione 60 ms, segnale di ingresso analogico 0÷10Vdc, 0÷20 mA o 4÷20 mA; Linea seriale in TTL, RS 232, RS 422, RS 485 o Current Loop; Possibilità di funzionamento in Idle-Mode o Power-Down Mode; Alimentatore switching, ad alto rendimento, incorporato; Alimentazione della sezione galvanicamente isolata: +24 Vdc; Alimentazione logica di bordo: 5 Vdc o 10÷40 Vdc o 8÷24Vac; Protezione alimentazione tramite TransZorb™; Firmware di telecontrollo, con protocollo di comunicazione ALB (**ABACO®** Link BUS); Vasta disponibilità di software di sviluppo quali Compilatore C, Assembler, BASIC Compiler BASCOM 8051, HTC-51, ecc.

grifo®
ITALIAN TECHNOLOGY
Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

GPC® R/T94 Edizione 3.40 *Rel. 25 Luglio 2001*

IMPORTANTE

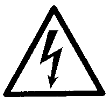
Tutte le informazioni contenute nel presente manuale sono state accuratamente verificate, ciononostante **grifo**® non si assume nessuna responsabilità per danni, diretti o indiretti, a cose e/o persone derivanti da errori, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.
grifo® altresi si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l' intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo per **grifo**®.
Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:



Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

Marchi Registrati

INDICE GENERALE

INDICE DELLE FIGURE

INTRODUZIONE

L'uso di questi dispositivi è rivolto - IN VIA ESCLUSIVA - a personale specializzato.

Scopo di questo manuale è la trasmissione delle informazioni necessarie all'uso competente e sicuro dei prodotti. Esse sono il frutto di un'elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

I dati di seguito riportati sono destinati - IN VIA ESCLUSIVA - ad un utenza specializzata, in grado di interagire con i prodotti in condizioni di sicurezza per le persone, per la macchina e per l'ambiente, interpretando un'elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall'ASSISTENZA TECNICA AUTORIZZATA, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

I dispositivi non possono essere utilizzati all'aperto. Si deve sempre provvedere ad inserire i moduli all'interno di un contenitore a norme di sicurezza che rispetti le vigenti normative. La protezione di questo contenitore non si deve limitare ai soli agenti atmosferici, bensì anche a quelli meccanici, elettrici, magnetici, ecc.

Per un corretto rapporto coi prodotti, è necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l'Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell'uso di tali dispositivi, è conveniente che l'utente - PRIMA DI COMINCIARE AD OPERARE - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

VERSIONI SCHEDE E FIRMWARE

Il presente manuale é riferito alle seguenti versioni:

- Scheda **GPC® R94**: versione **221199** e successive.
- Scheda **GPC® T94**: versione **221199** e successive.
- Firmware **ALB x94**: versione **1.3** e successive.

La validità delle informazioni riportate é quindi subordinata ai numeri di versione del sistema in uso e l'utente deve quindi sempre verificare la giusta corrispondenza tra le due indicazioni. Sulle schede il numero di versione é riportato in più punti sia a livello di serigrafia che di stampato (per esempio vicino alla batteria BT1 nel lato saldatura); la versione del firmware invece é riportata sul microprocessore, oppure può essere richiesta tramite l'apposito comando seriale.

INFORMAZIONI GENERALI

Le schede **GPC® R94** e **GPC® T94** (**G**eneral **P**urpose **C**ontroller **R**elays/**T**ransistors **9** input, **4** output) , che fanno parte della **Serie M** delle CPU con ingombro di 22,5x82x90 mm, sono dei potenti moduli di controllo, nella fascia **Low-Cost**, in grado di funzionare autonomamente come periferiche intelligenti e/o remotate in una più vasta rete di telecontrollo e/o di acquisizione.
La **GPC® R94** o **GPC® T94** é fornita di un contenitore in plastica provvisto degli attacchi per le guide ad **Omega** tipo **DIN 46277-1** e **DIN 46277-3**. Grazie al basso costo di questa serie di CPU é possibile affrontare proficuamente anche tutta quella piccola serie di automazioni che hanno un limitato badget di spesa. Sfruttando il ricco corredo di tools di sviluppo software, disponibili per la **GPC® R94** o **GPC® T94** come ad esempio il **BASCOM 8051** in abbinamento al **SIM2051**, é possibile effettuare le applicazioni in tempi sorprendentemente rapidi e con investimenti minimi.
La **GPC® R94** o **GPC® T94** é disponibile anche già programmata con un firmware di Telecontrollo. In questo caso, tramite il protocollo di comunicazione **ALB** (**ABACO® Link BUS**) può essere utilizzata come periferica intelligente, da sola o in rete, tramite una normale linea seriale. E' così possibile realizzare, in modo molto semplice, anche applicazioni, con un numero anche limitato di I/O, pilotate da un normale PC.
Le caratteristiche delle **GPC® R94** e **GPC® T94**, comprensive delle varie opzioni, sono le seguenti:

- Modulo Intelligente **ABACO® Block** della **Serie M**
- Ingombro di 22,5x82x90 mm
- Contenitore per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**
- **CPU 89C2051** o **89C4051**, con quarzo da **14MHz**
- **2K** o **4K Bytes** di **FLASH** e **128 Bytes** di **SRAM**
- Codice compatibile con i µP della famiglia **51**
- **Real Time Clock** con **256 Bytes** di **SRAM** interna
- Circuiteria di **Back-Up** per **RTC** e **SRAM**, tramite batteria al **LITIO**
- **E²** seriale da 1K Bytes
- **9** linee digitali, di ingresso, galvanicamente isolate
- Linea di **INT**, esterno, galvanicamente isolata
- **4** linee di uscita: A Relay da **5A** nella versione **R**
Optoisolote, di tipo **NPN** con Darlington da **4**A, **45** Vdc, privi di radiatore e con diodo di ricircolo nella versione **T**
- **LEDs**, di visualizzazione dello stato, per tutte le linee di I/O
- Collegamento delle I/O tramite connettori a rapida estrazione
- **2** linee di I/O TTL
- Registri di Timer-Counter da 16 bits
- Circuiteria per A/D converter con risoluzione di 11 bits. Tempo di conversione 60 ms. Segnale di ingresso analogico 0÷10Vdc, 0÷20 mA o 4÷20 mA
- Linea seriale in **TTL**, **RS 232**, **RS 422**, **RS 485** o **Current Loop**
- Possibilità di funzionamento in **Idle-Mode o Power-Down Mode**
- Alimentatore switching, ad alto rendimento, incorporato
- Alimentazione della sezione galvanicamente isolata: **+24 Vdc**
- Alimentazione logica di bordo: **5Vdc** o **10÷40 Vdc** o **8÷24Vac**
- Protezione alimentazione tramite **TransZorb™**
- Firmware di telecontrollo, con protocollo di comunicazione **ALB**
- Vasta disponibilità di software di sviluppo quali **Compilatore C**, **Assembler**, **BASIC Compiler BASCOM 8051**, **HTC-51**, ecc.

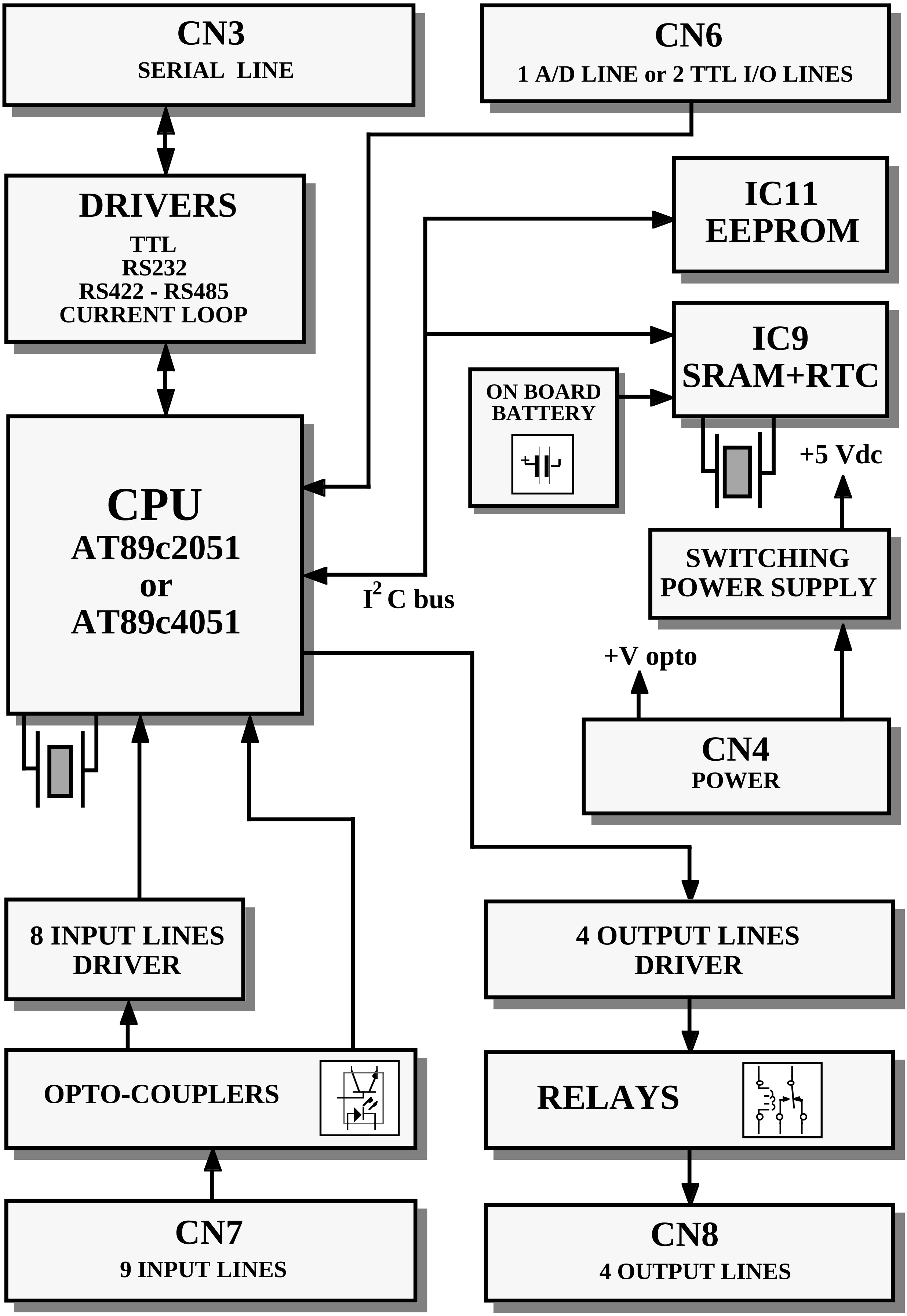


FIGURA 1: SCHEMA A BLOCCHI DELLA GPC® R94

Viene di seguito riportata una descrizione dei blocchi funzonali delle schede, con indicate le operazioni effettuate da ciascuno di essi. Per una più facile individuazione di tali blocchi e per una verifica delle loro connessioni, fare riferimento alle figure 1 e 2.

PROCESSORE DI BORDO

Le schede **GPC® R/T94** sono predisposte per accettare i processori ATMEL AT89C2051 o AT89C4051, con codice compatibile con la famiglia 51 INTEL. Tali processori ad 8 bits sono dotati di una ROM interna in formato FLASH, rispettivamente da 2 Kbytes e 4 Kbytes, per la programma applicativo, da un esteso set di istruzioni, da un'alta velocità di esecuzine e di manipolazione dati, da un efficiente gestione degli interrupts e da una ricca serie di periferiche hardware integrate.
Di seguito viene riportato un elenco delle caratteristiche principali delle CPU:

μP AT89C2051 AT89C4051:
- CPU ad 8 bit;
- 2 Kbytes o 4 Kbytes di ROM FLASH;
- 128 bytes di SRAM interna;
- 15 linee di I/O programmabili;
- 2 Timer/Counters da 16 bits;
- 5 sorgenti di interrupts con 2 livelli di priorità;
- 1 linea serialefull duplex;
- 1 comparatore analogico;
- Funzionamento in IDDLE-MODE o POWER-DOWN MODE.

Le **GPC® R/T94** vengono fornite di base, con un microprocessore AT89C4051 nel quale é programmato un firmware di telecontrollo, denominato **ALB x94**, in grado di interfacciarsi al protocollo **ABACO® Link BUS**. In questo caso é possibile anche il collegamento di più dispositivi, in modo da realizzare una rete di comunicazione e telecontrollo.
Per maggiori informazioni a riguardo di questo componente si faccia riferimento all'apposita documentazione della casa costruttrice, oppure all'appendice B di questo manuale.

SEZIONE ALIMENTATRICE

Le schede **GPC® R/T94** possono essere provviste di una efficiente sezione alimentatrice switching, che provvede a fornire la tensione di alimentazione di +5 Vdc, necessaria alle sezioni di logica e di output, in ogni condizione di carico e tensione d'ingresso; in assenza della sezione alimentatrice questa tensione deve essere fornita dall'esterno. Sulle schede sono state adottate tutte le scelte circuitali e componentistiche che tendono a ridurre i consumi, compresa la possibilità di far lavorare i microcontrollori in power-down ed iddle-mode ed a ridurre la sensibilità ai disturbi. Si ricorda inoltre che é presente una circuiteria di protezione tramite TransZorb™ per evitare danni dovuti a tensioni non corrette.
Per alimentare gli opto-isolatori delle sezioni galvanicamente isolate, é invece necessaria una tensione di 24 Vdc.

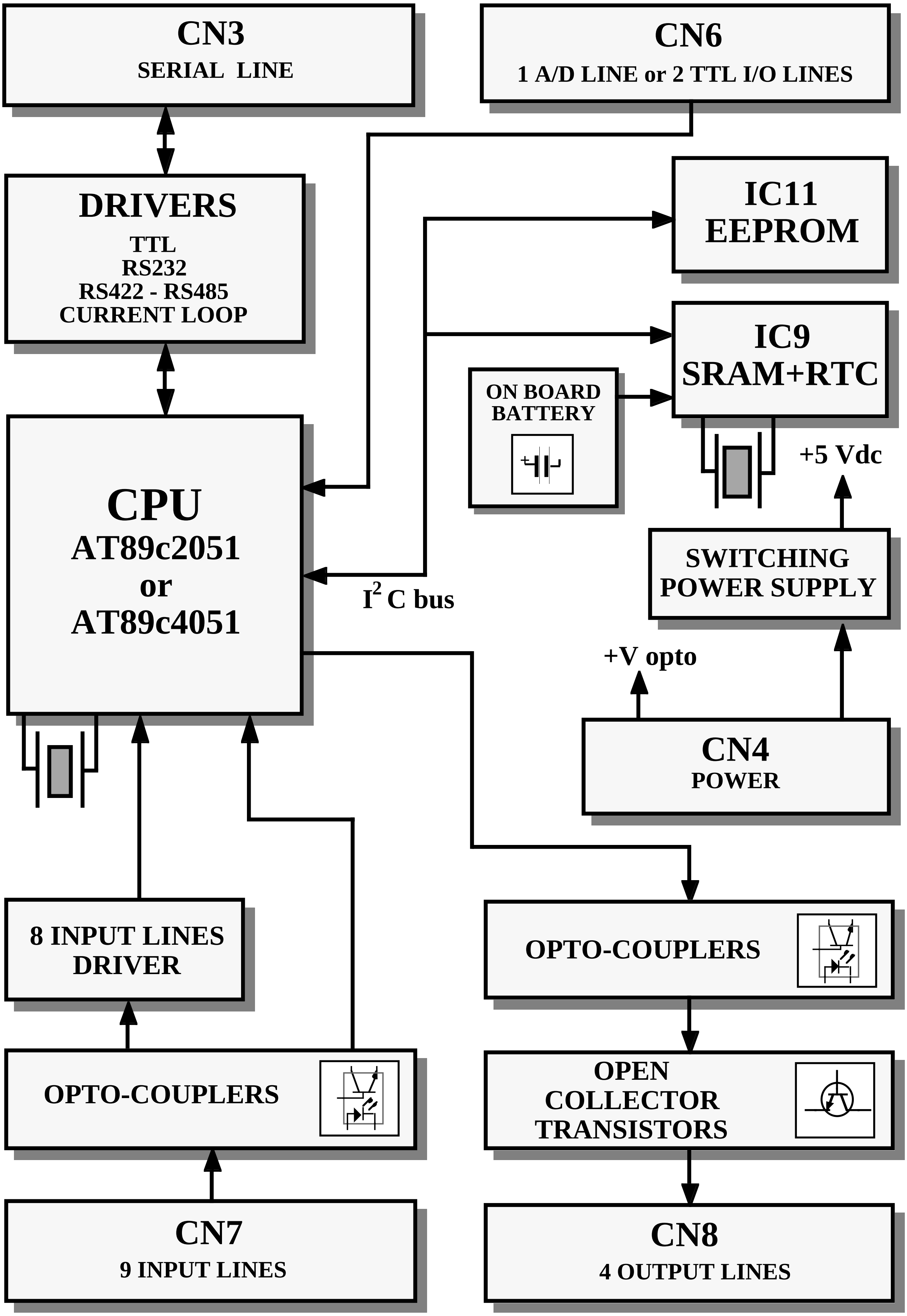


FIGURA 2: SCHEMA A BLOCCHI DELLA GPC® T94

CLOCK

Nelle schede **GPC® R/T94** vi sono due circuiterie separate, basate su altrettanti quarzi, per la generazione dei segnali di clock per il microprocessore (14.7456 MHz) e per il Real Time Clock (32.768 KHz). La scelta di utilizzare circuiterie di clock distinte per le varie sezioni delle schede, semplifica notevolmente l'uso delle stesse, infatti ad esempio può essere variata la frequenza di lavoro della CPU senza dover intervenire sul firmware di gestione delle altre sezioni.

MEMORIE E REAL TIME CLOCK

Le schede **GPC® R/T94** sono dotate di un Real Time Clock e di una quantità di memoria esterna che può arrivare fino a 1280 bytes. Questa é variamente suddivisa con un massimo di 256 bytes di SRAM statica e 1024 bytes di EEPROM seriale.
Tale caratteristica fornisce alla scheda la possibilità di ricordare in ogni condizione, una serie di parametri come ad esempio la configurazione o lo stato del sistema.

- **Real Time Clock**: il modulo di SRAM da 256 byte, montato su IC 9, é provvisto di un completo Real Time Clock in grado di gestire ore, minuti, secondi, giorno del mese, mese, anno e giorno della settimana in modo completamente autonomo. La scheda é dotata di una circuiteria che, tramite la batteria al litio di bordo, mantiene in back-up il componente anche in assenza della tensione di alimentazione.

- **EEPROM**: il modulo di EEPROM seriale da montato sul IC 11 é molto utile in caso si debbano mantenere delle informazioni anche in assenza di alimentazione, senza ricorrere al back up della SRAM, con una sicurezza estrema sulla validità dei dati. Tale modulo, nella configurazione standard, ha un size di 512 bytes, ma se esplicitamente ordinato può essere espanso fino a 1024 bytes.

Per ulteriori informazioni a riguardo dei dispositivi periferici descritti, si faccia riferimento alla documentazione tecnica della casa costruttrice.

SEZIONE DI OUTPUT A RELE'

La scheda **GPC® R94** é dotata di 4 uscite a relé da 5A, con contatto normale aperto, il cui stato é visualizzato da altrettanti LEDs. Ogni linea bufferata tramite un apposito driver, é pilotata direttamente dal microprocessore. Queste uscite sono disponibili su un comodo connettore a rapida estrazione, che permette un facile interfacciamento con i segnali del campo.

SEZIONE DI OUTPUT A TRANSISTORS

La scheda **GPC® T94** é dotata di 4 uscite a transistros Darlington NPN da 4A (non continuativi), collegati in open collector, con diodo di ricircolo, il cui stato é visualizzato da altrettanti LEDs. Ogni linea galvanicamente isolata tramite un apposito opto-isolatore, é pilotata direttamente dal microprocessore. Queste uscite sono disponibili su un comodo connettore a rapida estrazione, che permette un facile interfacciamento con i segnali del campo.

SEZIONE DI INPUT

Le schede **GPC® R/T94** sono dotate di 9 ingressi di tipo NPN, il cui stato é visualizzato da altrettanti LEDs. Le linee galvanicamente isolate, vengono acquisite dal microprocessore direttamente (1 associabile anche all'interrupt /INT0) oppure tramite un apposito buffer serializzatore (le restanti 8). Questi ingressi sono disponibili su un comodo connettore a rapida estrazione, che permette un facile interfacciamento con i segnali del campo.
Si ricorda che in caso di necessità é possibile acquisire solo 4 linee in maniera molto veloce eliminando il serializzatore e ponticellando opportunamente le linee necessarie. Per ulteriori informazioni contattare direttamente la **grifo®**.

A/D CONVERTER E LINEE DI I/O TTL

Le schede **GPC® R/T94** sono dotale di un ingresso analogico di conversione A/D, basato su una circuiteria che, tramite il comparatore analogico di bordo del microprocessore, é in grado di acquisire un segnale 0÷10 V, 0÷20 mA o 4÷20 mA, con una risoluzione di 11 bits ed un tempo di conversione massimo di 60 msec.
Questo ingresso analogico é disponibile su un comodo connettore a rapida estrazione, che permette un facile interfacciamento con il segnale del campo.
Inoltre, tramite una serie di jumpers, é possibile escludere tutta la circuiteria analogica e collegare i pin del microprocessore direttamente al connettore, in modo da avere disponibili 2 linee di I/O TTL generiche o gli ingressi del comparatore analogico senza nessuna circuiteria esterna.

COMUNICAZIONE SERIALE

Le schede **GPC® R/T94** dispongono di una linea seriale hardware completamente settabile via software, per quanto riguarda sia il protocollo sia la velocità di comunicazione. Tali settaggi avvengono tramite la programmazione dei registri interni al microcontrolloredi cui le schede sono provviste, quindi per ulteriori informazioni si faccia riferimento alla documentazione tecnica della casa costruttrice, oppure all'appendice B di questo manuale.
Dal punto di vista hardware, la linea seriale può essere configurata in TTL, RS232, current loop, RS 485 ed RS 422; in quest'ultimo caso é definibile anche se la comunicazione avviene in Full Duplex o Half Duplex.

FIRMWARE ALB X94

Le schede **GPC® R/T94** vengono fornite di base con il firmware **ALB x94** (**ABACO® L**ink **BUS**), che permette di gestire tutte le risorse della scheda tramite una serie di comandi sulla linea seriale. Una caratteristica interessante é che, sfruttando le linee di I/O TTL, sono stati implementati due distinti BUS di comunicazione per il protocollo **1-Wire®**; in questo modo, sempre tramite la linea seriale, possono essere gestiti fino a due dispositivi 1-Wire® collegati alla **GPC® R/T 94** (sensori di temperatura, memorie, Dallas iButton™, ecc.).
E' inoltre supportata una modalità di comunicazione Master-Slave che permette di remotare i singoli moduli anche a notevole disatanza, in modo da realizzare una rete di telecontrollo pilotata da un'unica unità master (PC, PLC, scheda della serie **GPC®**, ecc.).

SPECIFICHE TECNICHE GPC® R94

CARATTERISTICHE GENERALI GPC® R94

Risorse di bordo:	9 inputs digitali optoisolati NPN 4 outputs digitali a relé N.A. da 5 A 2 input/output TTL 1 interrupt esterno optoisolato NPN 1 comparatore analogico 1 linea di A/D converter 1 real time clock 1 linea seriale TTL, RS 232, RS 422-485 o Current Loop
Memoria indirizzabile:	IC 9: RTC+SRAM seriale da 256 bytes IC 11: EEPROM seriale da 512 bytes a 1 Kbytes
CPU di bordo:	Atmel AT89C2051 o AT89C4051
Frequenza clock :	14.7456 MHz (CPU) 32.768 KHz (real time clock)
Risoluzione A/D:	11 bits
Tempo conversione A/D:	Massimo 60 ms

CARATTERISTICHE FISICHE GPC® R94

Dimensioni (L x A x P):	111 x 90 x 22,5 mm 111 x 102 x 22,5 mm (incluso aggancio per guide Ω)
Montaggio:	Su guide Ω tipo DIN 46277-1 e DIN 46277-3
Peso:	182 g (versione base)
Connettori:	CN3: 4+4 vie AMP Mod II maschio 90° CN4: 4 vie rapida estrazione 90° CN6: 2 vie rapida estrazione 90° CN7: 10 vie rapida estrazione 90° CN8: 6 vie rapida estrazione 90°
Connettore femmina per CN3:	Codice **grifo®**: **CKS.AMP8** (kit composto da un AMP Mod II 4+4 vie femmina ed 8 contatti a crimpare)
Range di temperatura:	da 0 a 50 gradi Centigradi
Umidità relativa:	20% fino a 90% (senza condensa)

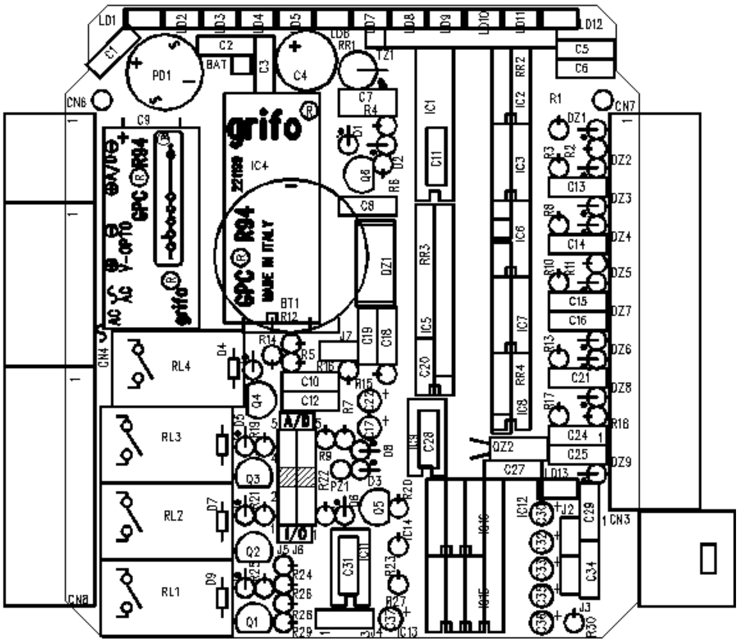


FIGURA 3: PIANTA COMPONENTI GPC® R94 (LATO COMPONENTI)

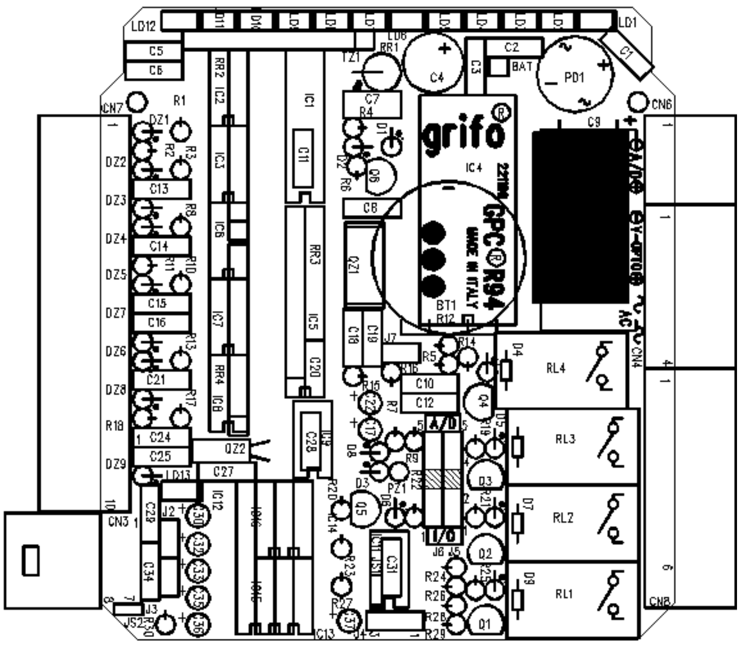


FIGURA 4: PIANTA COMPONENTI GPC® R94 (LATO SALDATURA)

CARATTERISTICHE ELETTRICHE GPC® R94

Versione senza alimentatore

Tensione di alimentazione:	+5 Vdc	(logica di controllo)
	+24 Vdc	(+V opto)

Versione con alimentatore switching

Tensione di alimentazione d'ingresso:	10÷40 Vdc o 8÷24 Vac	(logica di controllo)
	24 Vdc	(+V opto)
Tensione di alimentazione d'uscita:	+5 Vdc 740 mA	
	+Vopto 100 mA	
Corrente assorbita:	260 mA max	(+5 Vdc)
	40 mA max	(+V opto)
Corrente massima sul Relé:	5A	
Tensione massima sul Relé:	30 Vdc / 250 Vac	
Corrente minima per ingressi NPN:	2 mA	
Corrente di back up:	3,5 μA	
Batteria di bordo di back up:	3 V; 180 mAh; mod. CR 2032	
Ingresso analogico:	0÷10 V; 0÷20 mA; 4÷20 mA	
Impedenza ingresso analogico:	1 MΩ	
Rete di terminazione in RS422-485:	Resistenza di terminazione linea = 120 Ω	
	Resistenza di pull up sul positivo = 3,3 KΩ	
	Resistenza di pull down sul negativo = 3,3 KΩ	

CARATTERISTICHE GENERALI GPC® T94

Risorse di bordo:
9 inputs digitali optoisolati NPN
4 outputs digitali a transistors darlington NPN in open collector, con diodo di ricircolo
2 input/output TTL
1 interrupt esterno optoisolato NPN
1 comparatore analogico
1 linea di A/D converter
1 real time clock
1 linea seriale TTL, RS 232, RS 422-485 o Current Loop

Memoria indirizzabile:
IC 9: RTC+SRAM seriale da 256 bytes
IC 11: EEPROM seriale da 512 bytes a 1 Kbytes

CPU di bordo: Atmel AT89C2051 o AT89C4051

Frequenze clock CPU:
14.7456 MHz (CPU)
32.768 KHz (real time clock)

Risoluzione A/D: 11 bits

Tempo conversione A/D: Massimo 60 ms

CARATTERISTICHE FISICHE GPC® T94

Dimensioni (L x A x P):
111x 90 x 22,5 mm
111 x 102 x 22,5 mm (incluso aggancio per guide Ω)

Montaggio: Su guide Ω tipo DIN 46277-1 e DIN 46277-3

Peso: 171 g (versione base)

Connettori:
CN3: 4+4 vie AMP Mod II maschio 90°
CN4: 4 vie rapida estrazione 90°
CN6: 2 vie rapida estrazione 90°
CN7: 10 vie rapida estrazione 90°
CN8: 6 vie rapida estrazione 90°

Connettore femmina per CN3: Codice **grifo®**: **CKS.AMP8** (kit composto da un AMP Mod II 4+4 vie femmina ed 8 contatti a crimpare)

Range di temperatura: da 0 a 50 gradi Centigradi

Umidità relativa: 20% fino a 90% (senza condensa)

CARATTERISTICHE ELETTRICHE GPC® T94

Versione senza alimentatore

Tensione di alimentazione:	+5 Vdc	(logica di controllo)
	+24 Vdc	(+V opto)

Versione con alimentatore switching

Tensione di alimentazione d'ingresso:	10÷40 Vdc o 8÷24 Vac	(logica di controllo)
	24 Vdc	(+V opto)
Tensione di alimentazione d'uscita:	+5 Vdc; 930 mA	
	+Vopto 100 mA	
Corrente assorbita:	70 mA max	(+5 Vdc)
	40 mA max	(+V opto)
Corrente massima sul Transistor:	4A non continuativi	(*)
Tensione massima sul Transistor:	45 Vdc	(*)
Potenza massima sul Transistor:	1,25 W	(*)
Corrente minima per ingressi NPN:	2 mA	
Corrente di back up:	3,5 μA	
Batteria di bordo di back up:	3 V; 180 mAh; mod. CR 2032	
Ingresso analogico:	0÷10 V; 0÷20 mA; 4÷20 mA	
Impedenza ingresso analogico:	1 MΩ	

Rete di terminazione in RS422-485:	Resistenza di terminazione linea	=	120 Ω
	Resistenza di pull up sul positivo	=	3,3 KΩ
	Resistenza di pull down sul negativo	=	3,3 KΩ

(*) I valori sono riferiti ad una temperatura di lavoro di 20 °C

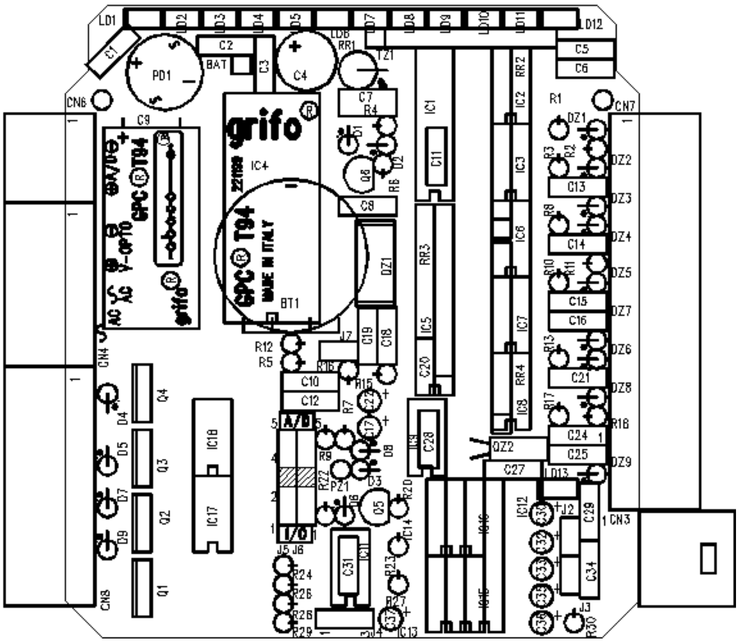


FIGURA 5: PIANTA COMPONENTI GPC® T94 (LATO COMPONENTI)

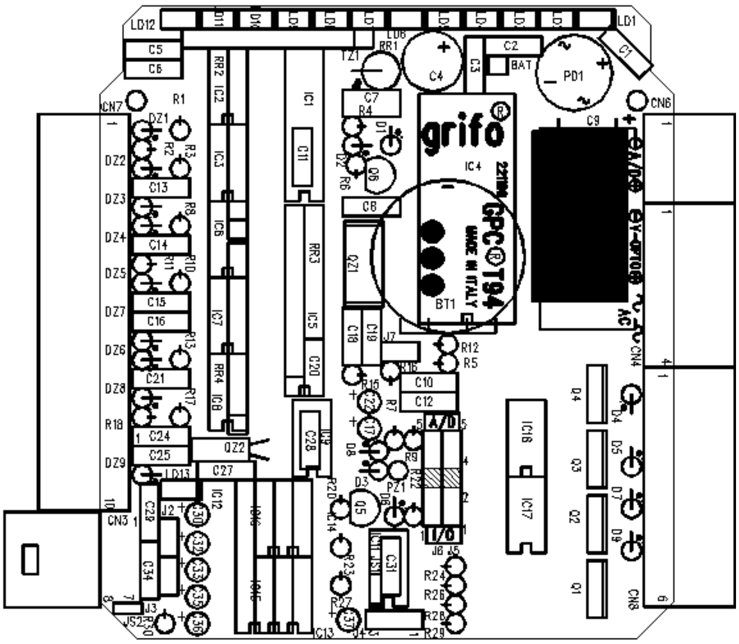


FIGURA 6: PIANTA COMPONENTI GPC® T94 (LATO SALDATURA)

INSTALLAZIONE

In questo capitolo saranno illustrate tutte le operazioni da effettuare per il corretto utilizzo delle schede. A questo scopo viene riportata l'ubicazione e la funzione degli strip, dei connettori, dei LEDs, ecc. presenti sulle **GPC® R/T94**.

CONNESSIONI CON IL MONDO ESTERNO

Le schede **GPC® R/T94** sono provviste di 5 connettori con cui vengono effettuate tutte le connessioni con il campo e con le altre schede del sistema di controllo da realizzare. Di seguito viene riportato il loro pin out ed il significato dei segnali collegati; per una facile individuazione di tali connettori, si faccia riferimento alle figure 8 e 9, mentre per ulteriori informazioni a riguardo del tipo di connessioni, fare riferimento alle figure successive che illustrano il tipo di collegamento a bordo scheda.

CN4 - CONNETTORE DI ALIMENTAZIONE

CN4 é un connettore a morsettiera a rapida estrazione, composto da 4 contatti.
Tramite CN4 devono essere fornite le tensioni di alimentazione, necessarie all'alimentatore switching di bordo ed agli opto-isolatori della seziona di input NPN.
Si ricorda che per un corretto funzionamento delle schede, queste due tensioni devono essere galvanicamente isolate tra di loro.

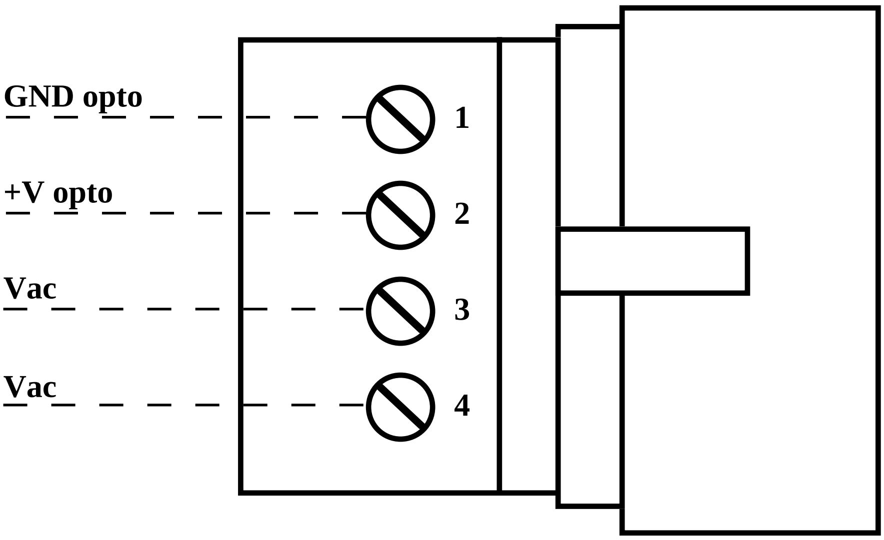


FIGURA 7: CN4 - CONNETTORE DI ALIMENTAZIONE

Legenda:

+V opto = I - Positivo della tensione di alimentazione +V opto
GND opto = - Linea di massa della tensione di alimentazione +V opto.
Vac = I - Linea di alimentazione per logica di controllo (+10÷40 Vdc o 8÷24Vac).
Vac = I - Linea di alimentazione per logica di controllo (+10÷40 Vdc o 8÷24Vac).

Per maggiori informazioni vedere il paragrafo "TENSIONI DI ALIMENTAZIONE".

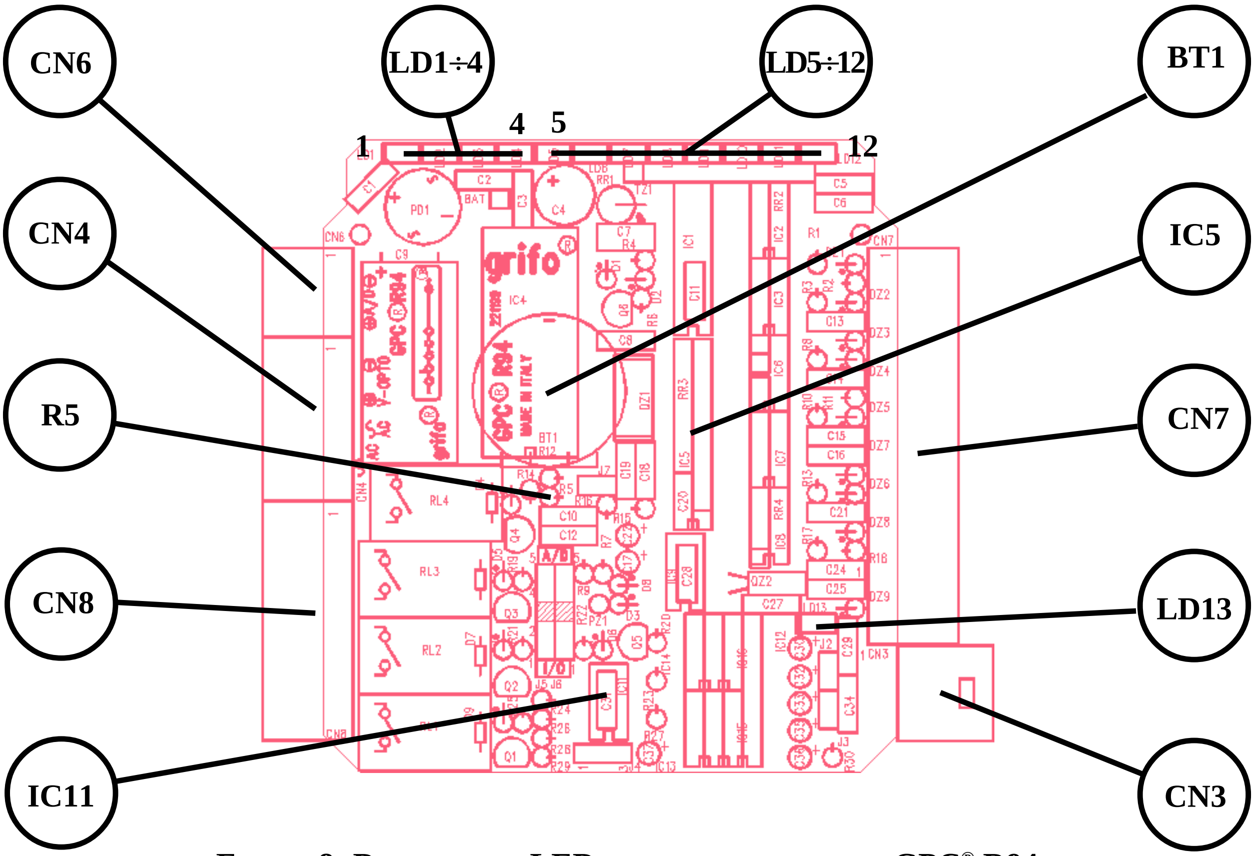


FIGURA 8: DISPOSIZIONE LEDS, CONNETTORI, ECC. SU GPC® R94

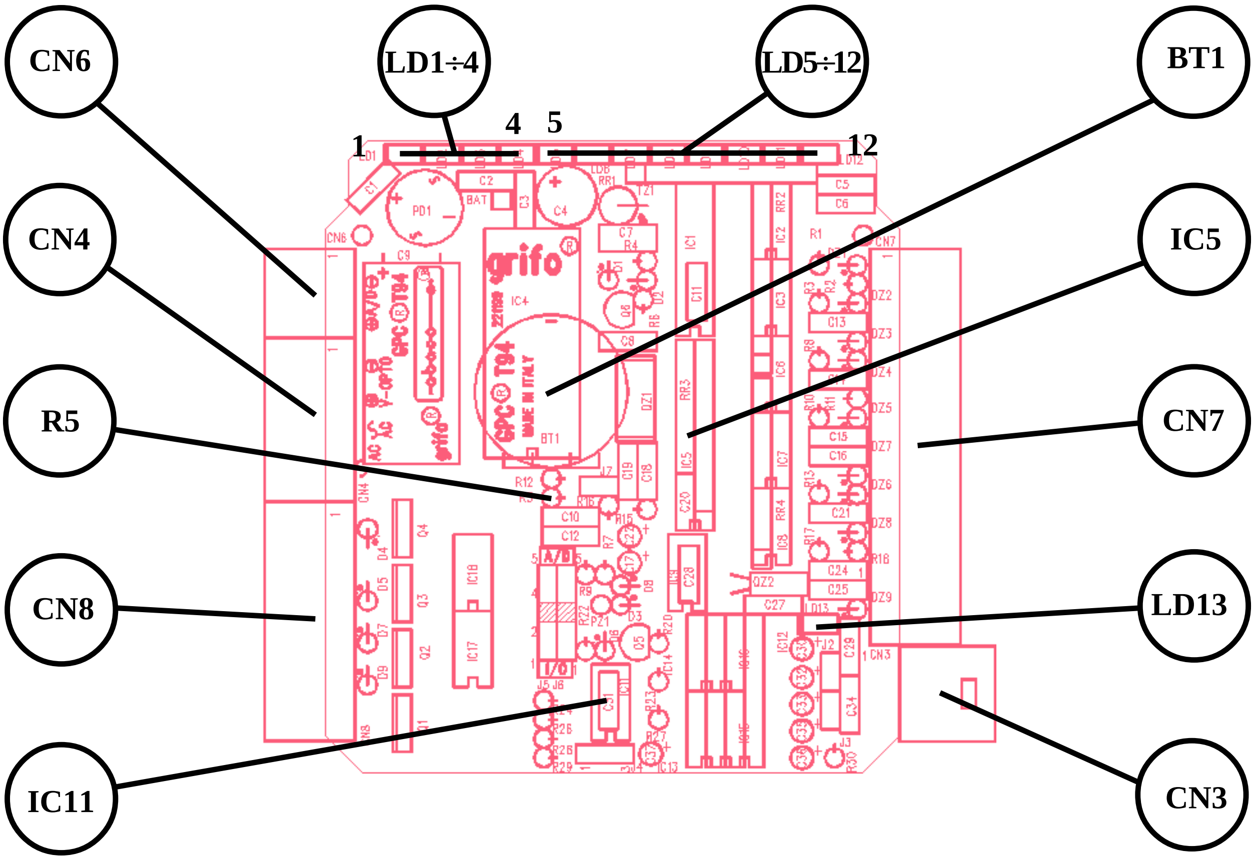


FIGURA 9: DISPOSIZIONE LEDS, CONNETTORI, ECC. SU GPC® T94

CN3 - CONNETTORE PER LINEA SERIALE ED ALIMENTAZIONE

CN3 é un connettore del tipo AMP Mod II maschio, a 4+4 vie con passo 2.54 mm.
Sul connettore CN3 sono disponibili la linea seriale e le tensioni di alimentazione. Se le schede sono dotate di alimentatore switching, tramite CN3 l'utilizzatore può alimentare dei carichi esterni prelevando le due tensioni galvanicamente isolate. Viceversa (alimentatore non presente), tramite CN3 é possibile fornire le due tensioni (+5 Vdc e +V opto). La disposizione dei pin, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo; inoltre i segnali della linea seriale rispettano le normative definite dal CCITT relative ad ognuno degli standard di comunicazione usati.
Il connettore femmina per CN3 può essere ordinato alla **grifo®** (codice **CKS.AMP8**), mentre acquistando direttamente dal catalogo AMP, fare riseferimento ai seguenti P/N: 280365 (connettore AMP Mod II femmina 4+4 vie) e 182206-2 (contatti a crimpare).

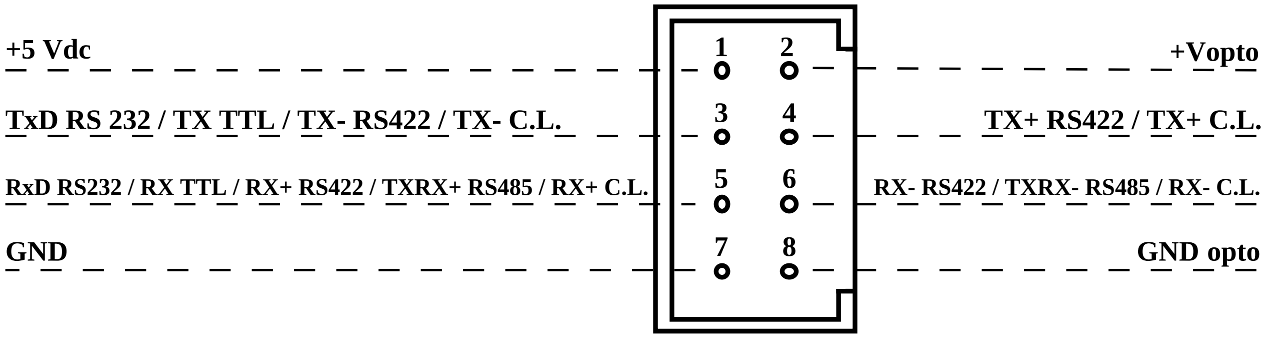


FIGURA 10: CN3 - CONNETTORE PER LINEA SERIALE ED ALIMENTAZIONE

Legenda:

RX TTL	=	I	- Receive Data: linea di ricezione della linea seriale in TTL.
TX TTL	=	O	- Transmit Data: linea di trasmissione della linea seriale in TTL.
RxD RS 232	=	I	- Receive Data: linea di ricezione della linea seriale in RS 232.
TxD RS 232	=	O	- Transmit Data: linea di trasmissione della linea seriale in RS 232.
RX- RS 422	=	I	- Receive Data Negative: linea bipolare negativa di ricezione differenziale della linea seriale in RS 422.
RX+ RS 422	=	I	- Receive Data Positive: linea bipolare positiva di ricezione differenziale della linea seriale in RS 422.
TX- RS 422	=	O	- Transmit Data Negative: linea bipolare negativa di trasmissione differenziale della linea seriale in RS 422.
TX+ RS 422	=	O	- Transmit Data Positive: linea bipolare positiva di trasmissione differenziale della linea seriale in RS 422.
TXRX- RS 485	=	I/O	- Transmit Receive Data Negative: linea bipolare negativa di ricezione e trasmissione differenziale della linea seriale in RS 485.
TXRX+ RS 485	=	I/O	- Transmit Receive Data Positive: linea bipolare positiva di ricezione e trasmissione differenziale della linea seriale in RS 485.
RX- C.L.	=	I	- Receive Data Negative: linea bipolare negativa di ricezione in Current Loop della linea seriale.
RX+ C.L.	=	I	- Receive Data Positive: linea bipolare positiva di ricezione in Current Loop della linea seriale.
TX- C.L.	=	O	- Transmit Data Negative: linea bipolare negativa di trasmissione in Current Loop della linea seriale.

TX+ C.L.	=	O	- Transmit Data Positive: linea bipolare positiva di trasmissione in Current Loop della linea seriale.
+5 Vdc	=	I/O	- Positivo della tensione di alimentazione a +5 Vdc.
GND	=		- Linea di massa digitale.
+V opto	=	I/O	- Positivo della tensione di alimentazione +V opto
GND opto	=		- Linea di massa della tensione di alimentazione +V opto.

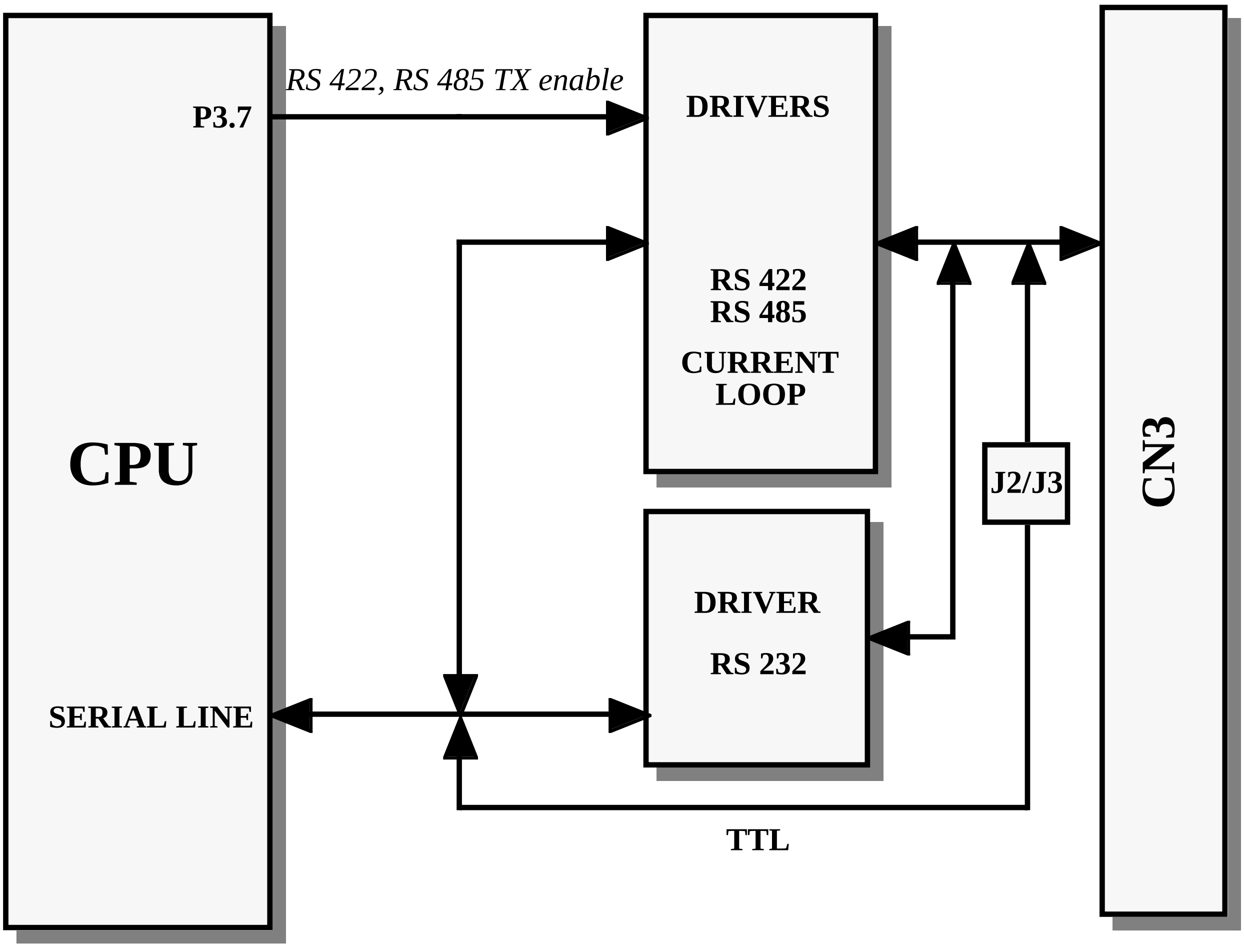


FIGURA 11: SCHEMA DI COMUNICAZIONE SERIALE

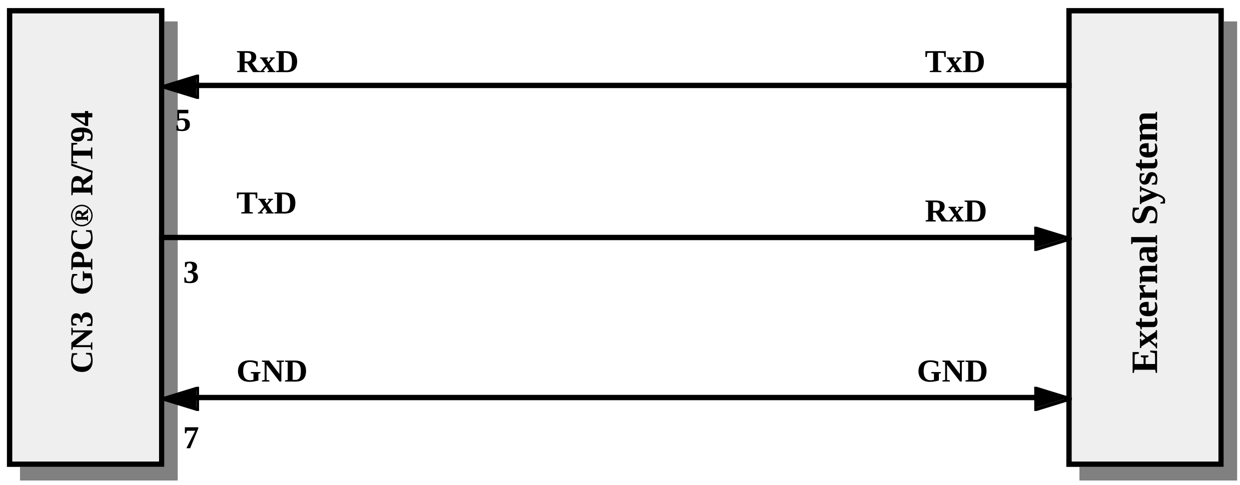


FIGURA 12: ESEMPIO DI COLLEGAMENTO PUNTO PUNTO IN RS 232

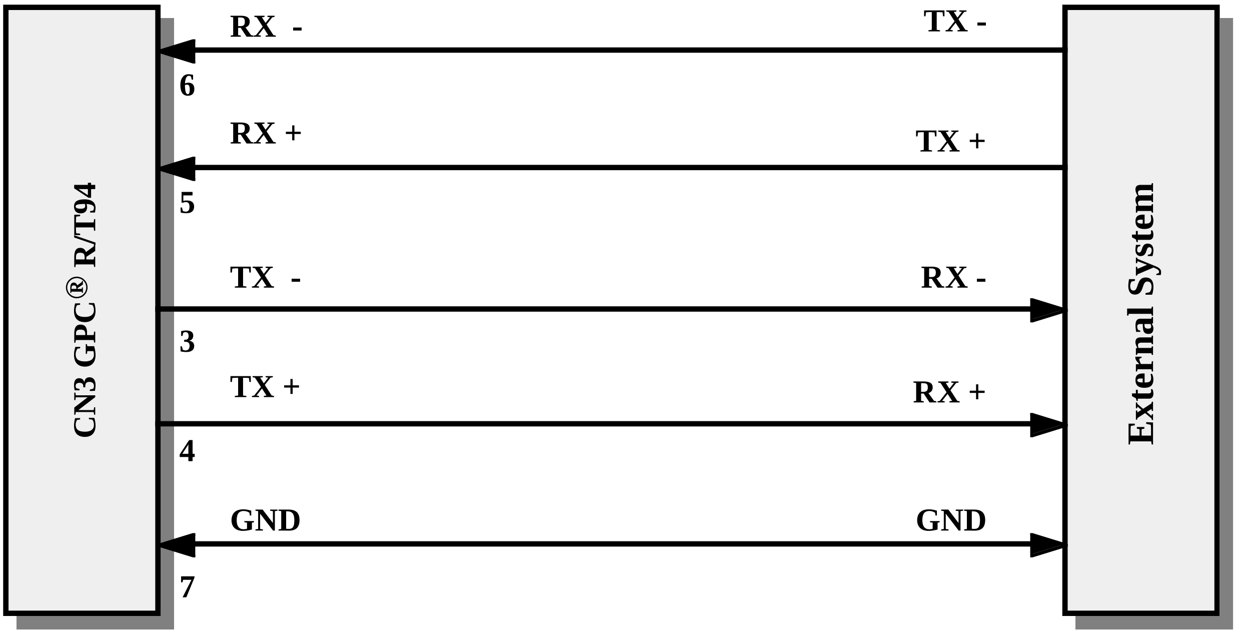


FIGURA 13: ESEMPIO DI COLLEGAMENTO PUNTO PUNTO IN RS 422

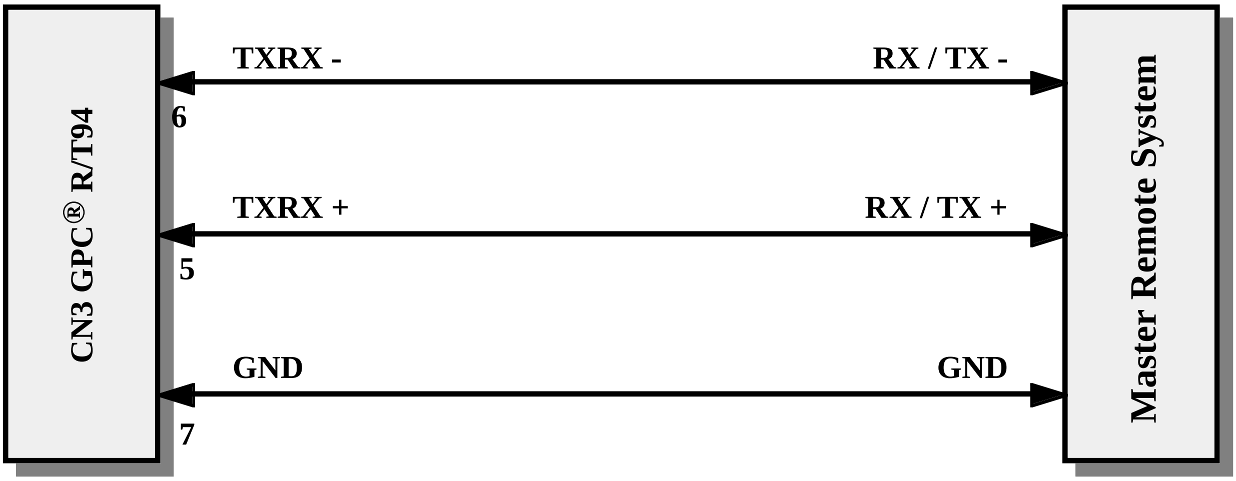


FIGURA 14: ESEMPIO DI COLLEGAMENTO PUNTO PUNTO IN RS 485

NOTA BENE

Con il collegamento in RS 485, se si usa il firmware **ALB x94**, é possibile utilizzare solo la modalità di comunicazione Master-Slave a 9 bit.

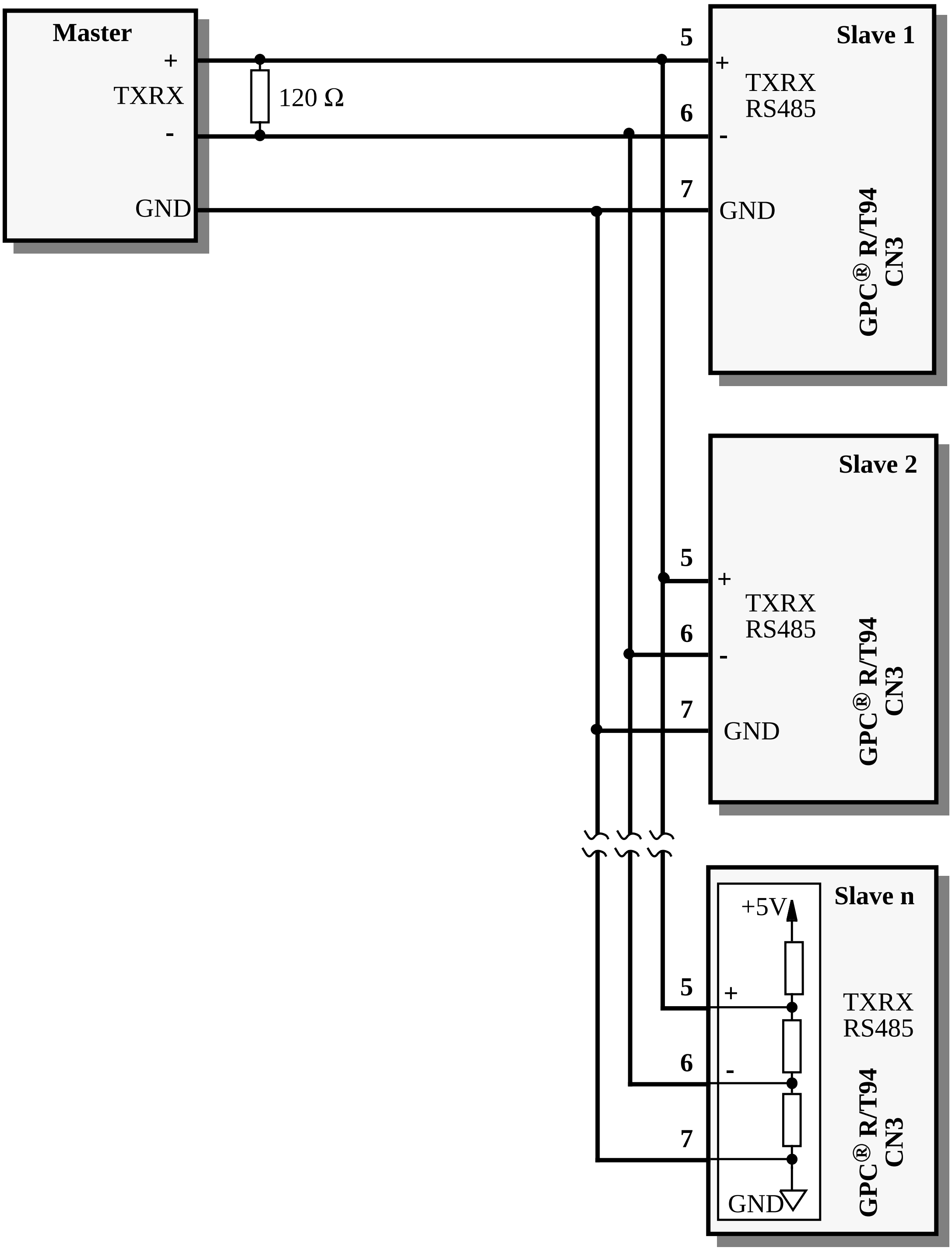


FIGURA 15: ESEMPIO DI COLLEGAMENTO IN RETE RS 485

Da notare che in una rete RS 485, devono essere presenti due resistenze di forzatura lungo la linea e due resitenze di terminazione (120 Ω), alle estremità della stessa, rispettivamente vicino all'unità Master ed all'ultima unità Slave.
A bordo delle **GPC® R/T94** è presente la circuiteria di terminazione e forzatura, che può essere inserita o disinserita, tramite appositi jumpers, come illutrato in seguito.
In merito alla resistenza di terminazione dell'unità Master, provvedere a collegarla solo se questa non é già presente al suo interno (ad esempio molti convertitori RS232-RS485 ne sono già provvisti).
Per maggiori informazioni consultare il Data-Book TEXAS INSTRUMENTS, "*RS 422 and RS 485 Interface Cicuits*", nella parte introduttiva riguardante le reti RS 422-485.

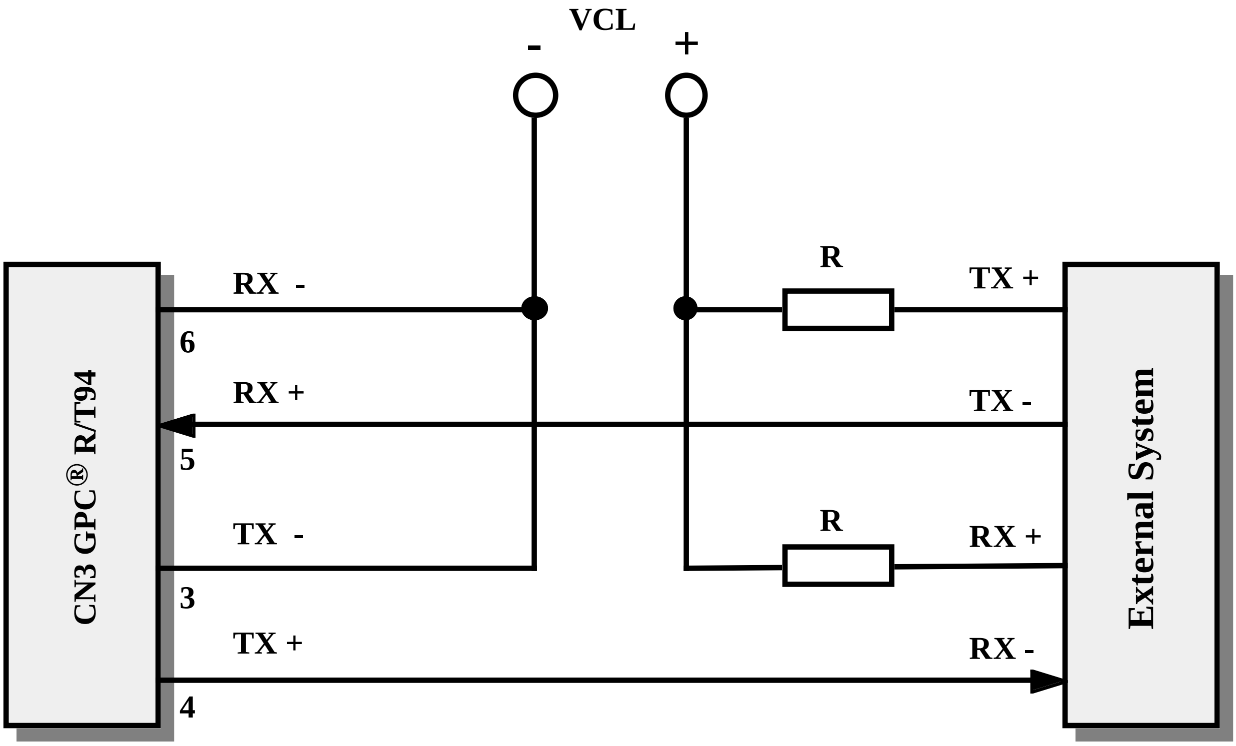


FIGURA 16: ESEMPIO DI COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI

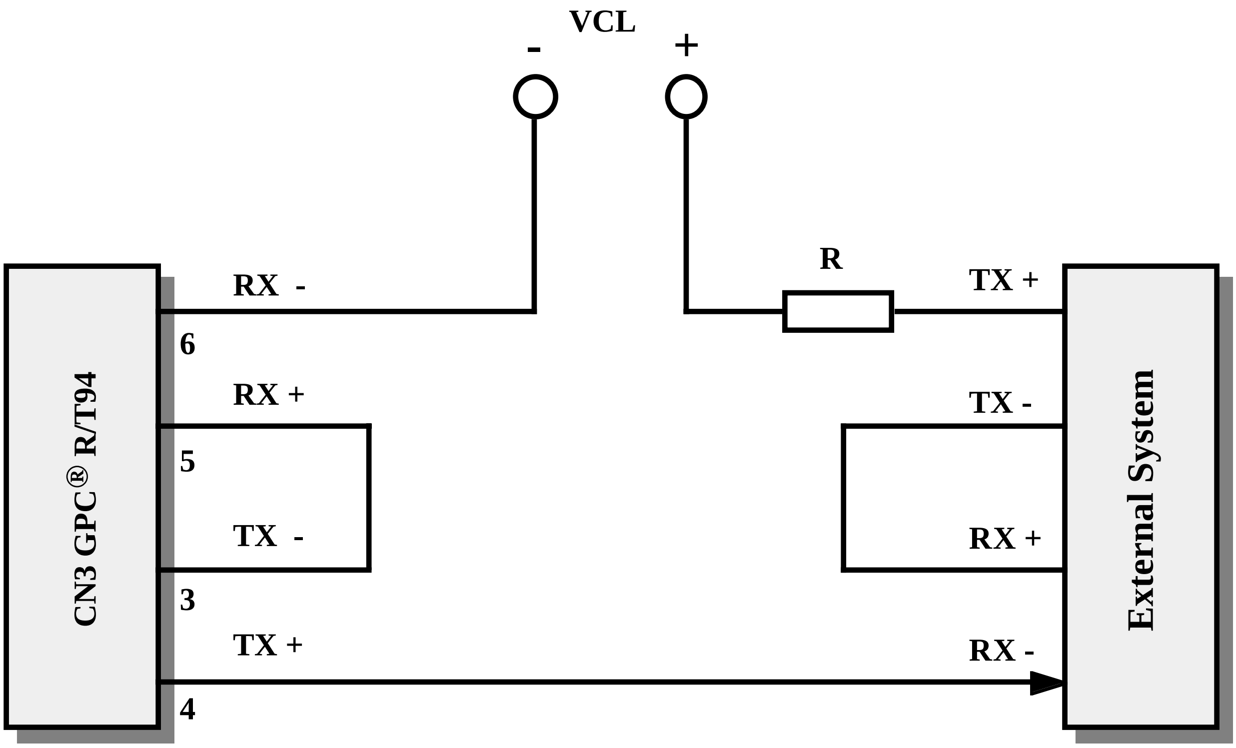


FIGURA 17: ESEMPIO DI COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI

NOTA BENE

Con il collegamento in current loop a 2 fili, se si usa il firmware **ALB x94**, é possibile utilizzare solo la modalità di comunicazione Master-Slave a 9 bit.

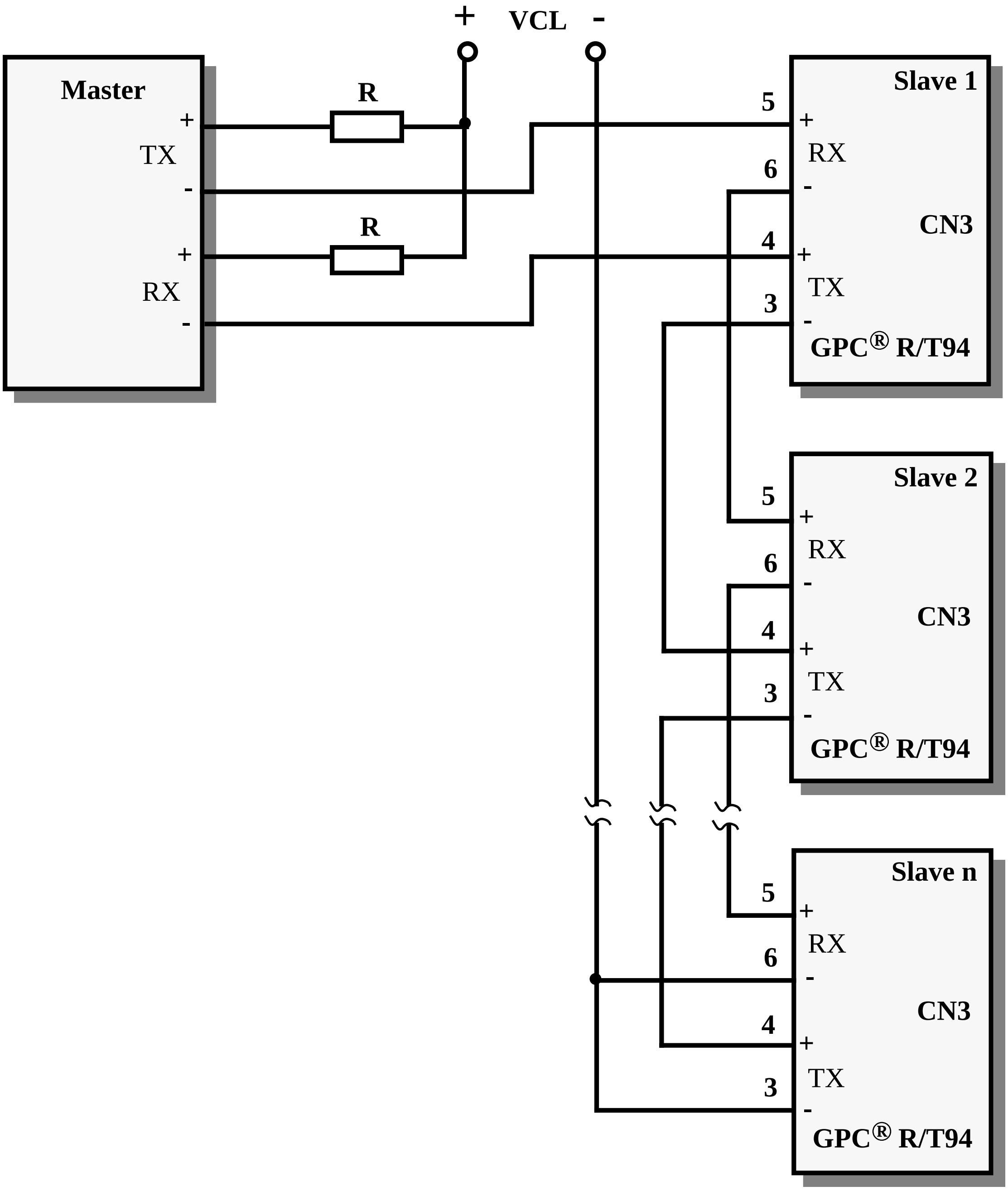


FIGURA 18: ESEMPIO DI COLLEGAMENTO IN RETE IN CURRENT LOOP PASSIVO

Per il collegamento in Current Loop passivo sono possibili due diversi tipi di collegamento: a 2 fili ed a 4 fili. Tali connessioni sono riportate nelle figure 19 e 20; in esse é indicata la tensione per alimentare l'anello (**VCL**) e le resistenze di limitazione della corrente (**R**). I valori di tali componenti variano in funzione del numero di dispositivi collegati e della caduta sul cavo di collegamento; bisogna quindi effettuare la scelta considerando che:

- si deve garantire la circolazione di una corrente di **20 mA**;
- su ogni trasmettitore cadono mediamente **2,35 V** con una corrente di 20 mA;
- su ogni ricevitore cadono mediamente **2,52 V** con una corrente di 20 mA;
- in caso di cortocircuito sulla rete ogni trasmettitore dissipi al massimo **125 mW**;
- in caso di cortocircuito sulla rete ogni ricevitore dissipi al massimo **90 mW**.

Per maggiori informazioni consultare il Data-Book HEWLETT-PACKARD, nella parte che riguarda gli opto-accoppiatori per Current Loop denominati **HCPL 4100** e **HCPL 4200**.

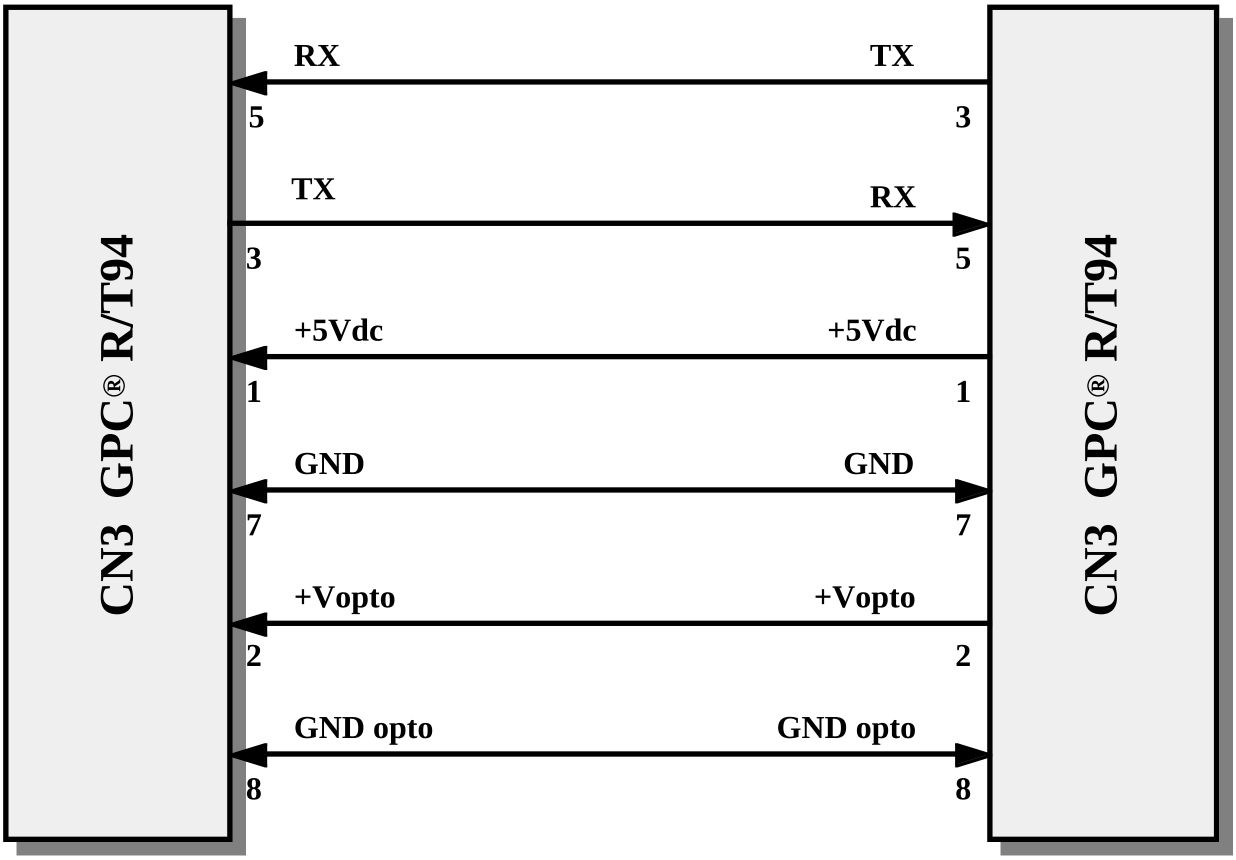


FIGURA 19: ESEMPIO DI COLLEGAMENTO SERIALE TTL ED ALIMENTAZIONE

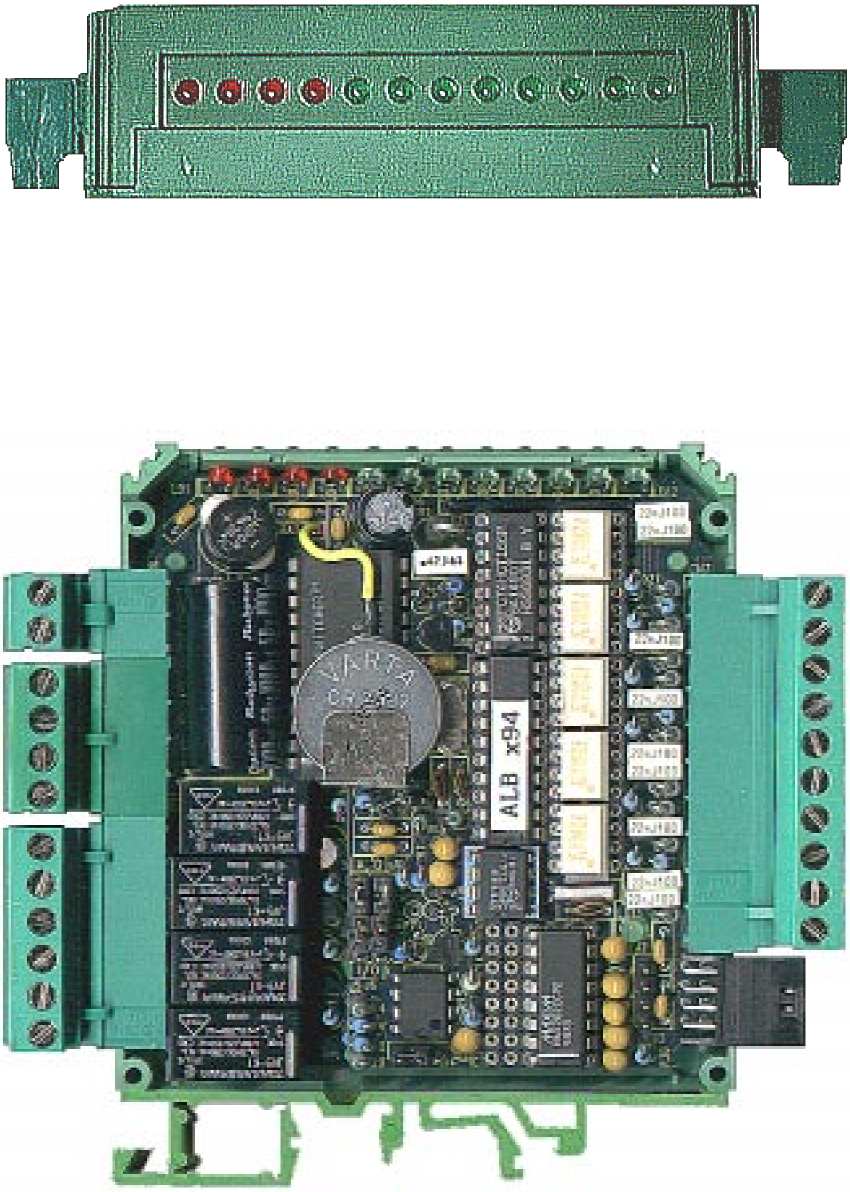


FIGURA 20: FOTO DELLA GPC® R94

CN7 - CONNETTORE PER INGRESSI OPTOISOLATI

CN7 é un connettore a morsettiera per rapida estrazione, composto da 10 contatti.
Tramite CN7 possono essere collegati i 9 input optoisolati di tipo NPN, disponibili sulle schede **GPC® R/T94**; in particolare gli 8 ingressi connessi al driver serializzatore 74 HCT 166 ed il restante collegato direttamente al microprocessore.
Sul connettore oltre alle linee degli ingressi, é presente anche il segnale di massa dell' alimentazione dei relativi optoisolatori.

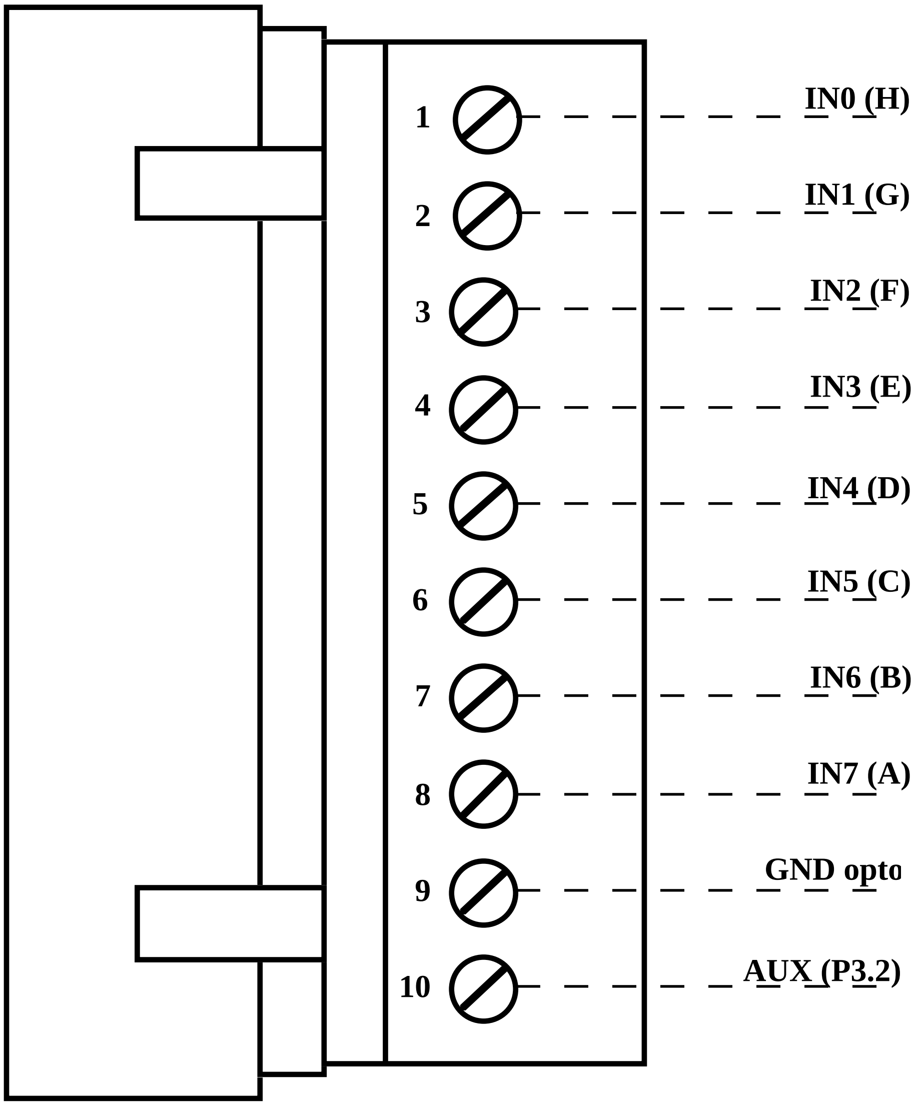


FIGURA 21: CN7 - CONNETTORE PER INGRESSI OPTOISOLATI

Legenda:

IN0÷IN7	=	I - Ingresso opto-isolato NPN, connessi ai pin H÷A del driver serializzatore 74 HCT 166.
AUX	=	I - Ingresso opto-isolato NPN, connesso al pin P3.2 (/INT0) della CPU.
GND opto	=	- Linea di massa della tensione di alimentazione +V opto

Le linee di input disponibili sulle schede, sono del tipo optoisolato e sono dotati di filtro passa-basso; in questo modo é garantita una certa protezione dell'elettronica interna, rispetto ai possibili disturbi provenienti dall'esterno. Ogni linea comprende un diodo LED con funzione di feed-back visivo (il LED si accenderà tutte le volte in cui l'ingresso risulterà portato alla GND opto); gli ingressi supporteranno, quindi, contatti normalmente aperti. In particolare tali linee sono adatte a driver del tipo **NPN**. Nel caso si debbano collegare a driver del tipo **PNP** si deve interporre un modulo della serie Block tipo **PBI 01**.
La circuiteria di una sezione di Input, composta da 9 linee, é rappresentata nel seguente schema.
Per quanto riguarda la tensione di alimentazione degli optoisolatori, questa deve essere fornita tramite l'apposito connettore CN4.

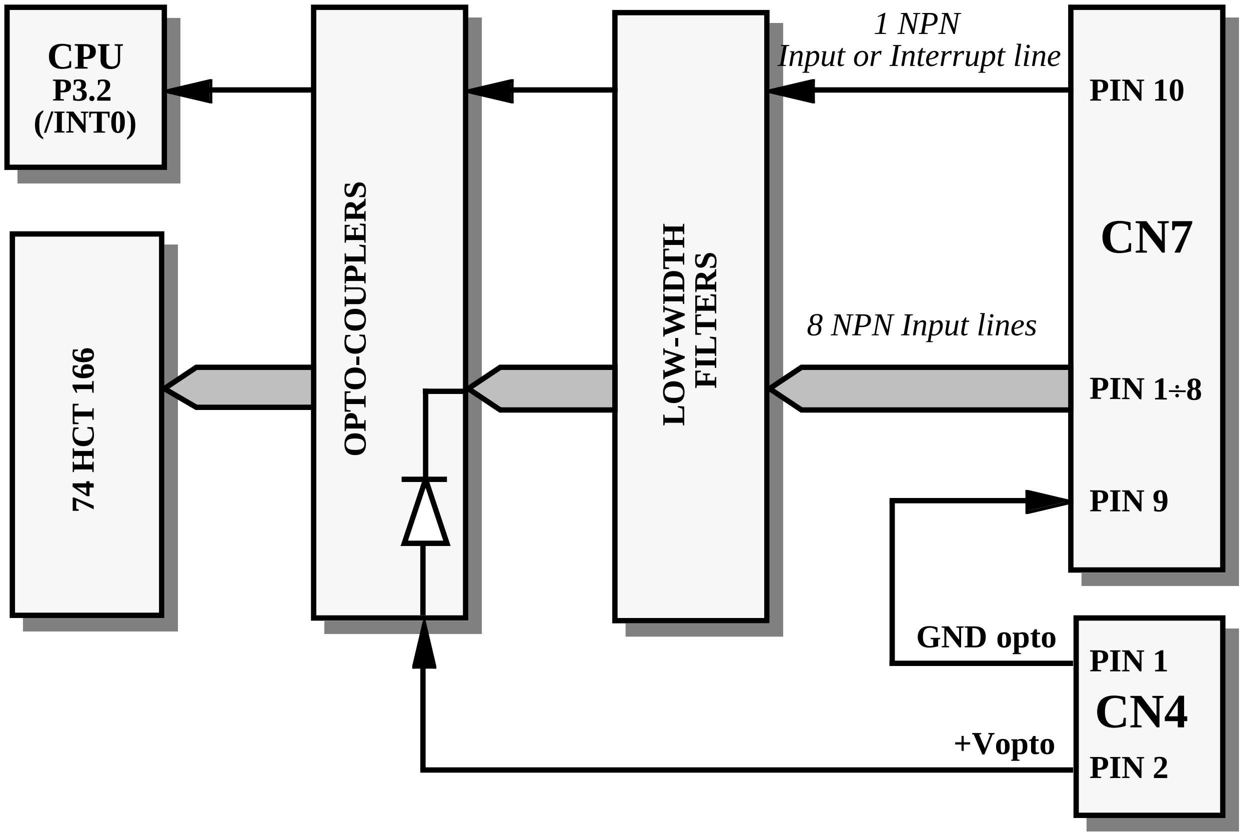


FIGURA 22: SCHEMA DEGLI INGRESSI OPTOISOLATI

CN8 - CONNETTORE PER USCITE A RELE’

CN8 é un connettore a morsettiera per rapida estrazione, composto da 6 contatti.
Tramite CN8 possono essere collegati i contatti normali aperti ed i relativi comuni delle 4 uscite a relé, presenti sulla **GPC® R94**. In fase di collegamento si deve ricordare che il carico massimo sopportato da ogni linea è di **5 A** con un tensione massima di **30 Vdc** o **250 Vac**.
La gestione di queste uscite avviene tramite una serie di pins di I/O del microcontrollore, opportunamente bufferati, i quali sono stati accuratamente scelti, in modo da semplificare al massimo la gestione software (per maggiori informazioni vedere il capitolo “DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO”).

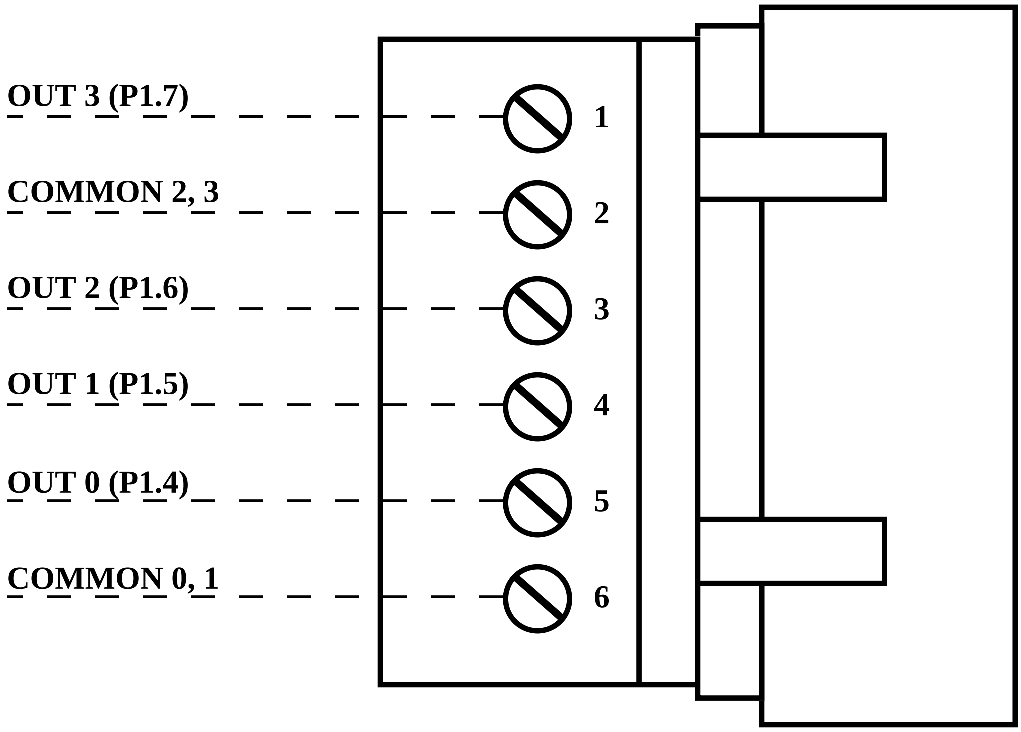


FIGURA 23: CN8 - CONNETTORE PER USCITE A RELÈ

Legenda:

OUT n (Px.y) = O - Contatto normale aperto del relé n, pilotato dal pin Px.y del microcontrollore.
COMMON n, m = - Contatto comune dei relé n ed m.

Le linee di output a relé, comprendono un diodo LED con funzione di feed-back visivo (il LED si accenderà tutte le volte in cui il contatto del relé risulterà chiuso). I relè sono pilotati da 4 transistrors PNP che a loro volta sono gestiti attraverso altrettanti pins di I/O del microcontrollore.

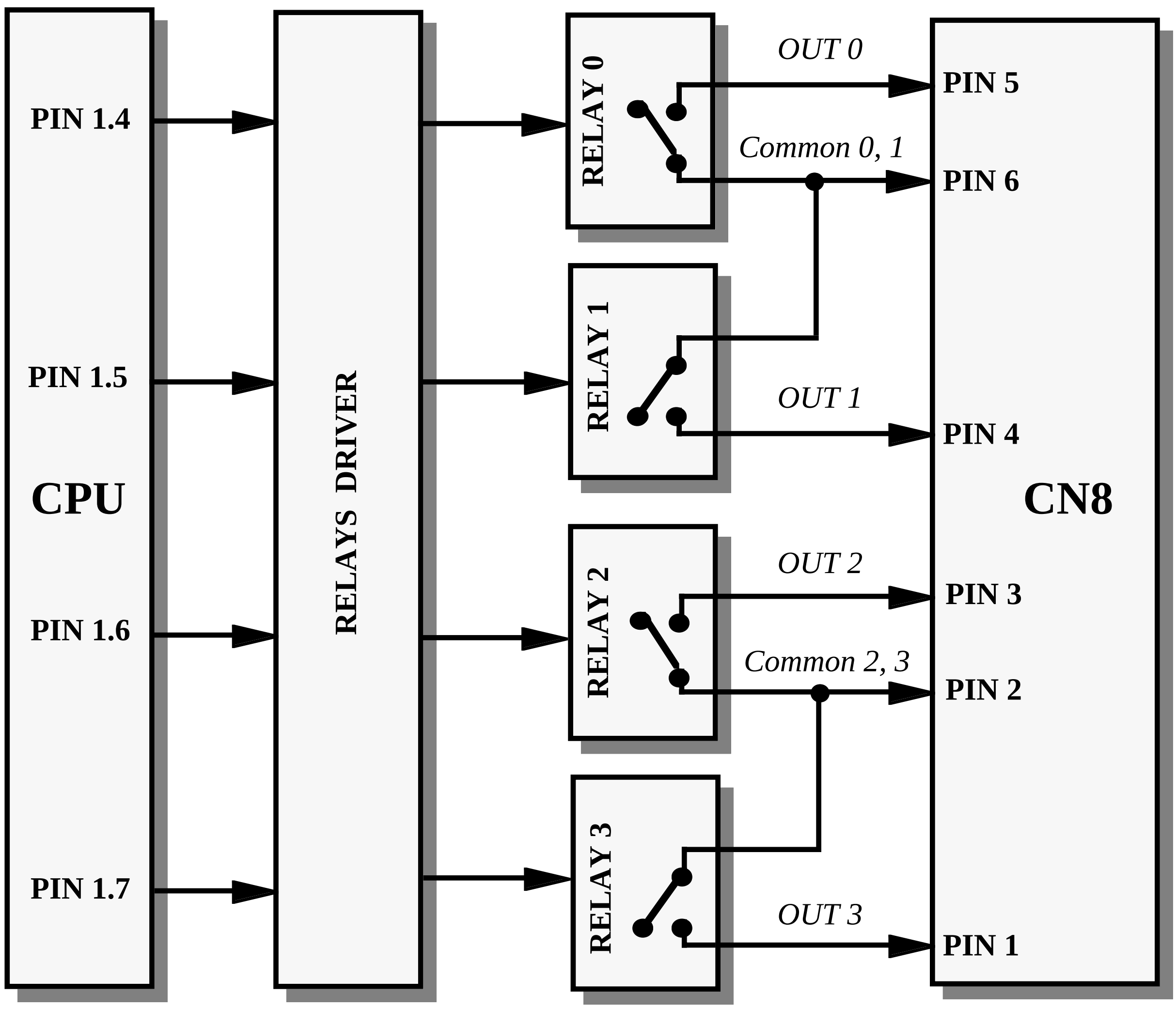


FIGURA 24: SCHEMA DELLE USCITE A RELÉ

CN8 - CONNETTORE PER USCITE A TRANSISTORS

CN8 é un connettore a morsettiera per rapida estrazione, composto da 6 contatti.
Tramite CN8 possono essere collegati i segnali in open collector ed il relativo comune (emitter) delle 4 uscite a a transistor darlington NPN, presenti sulla **GPC® T94**. In fase di collegamento si deve ricordare che il carico massimo sopportato da ogni linea è di **4 A** non continuativi, con un tensione massima di **45 Vdc**. Da notare che i transistors, essendo privi di radiatore, sono in grado di pilotare in maniera continuativa, un carico resistivo che, alimentato a **24 Vdc**, assorbe una corrente massima di **600 mA**, questo a condizione che la temperatura di lavoro rimanga a 20 gradi centigradi.
Tutte le linee sono dotare di un diodo di ricircolo, il quale elimina eventuali tensione induttive, create dall'attivazione dell'uscita, quando vengono pilotati carichi come relé di potenza, solenoidi, elettro-valvole, ecc. In questo caso é necessario collegare l'alimentazione del carico al segnale +VL.
La gestione delle uscite avviene tramite una serie di pins di I/O del microcontrollore, opportunamente bufferati, i quali sono stati accuratamente scelti, in modo da semplificare al massimo la gestione software (per maggiori informazioni vedere il capitolo "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO").

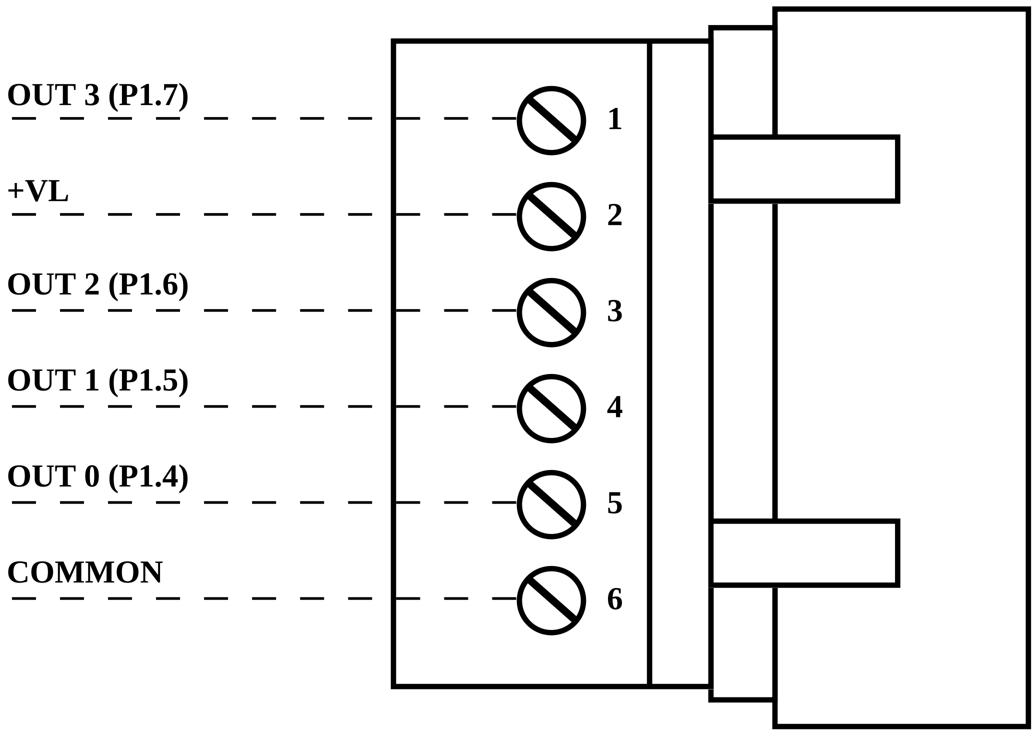


FIGURA 25: CN8 - CONNETTORE PER USCITE A TRANSISTORS

Legenda:

OUT n (Px.y)	= O	- Contatto in open collector del transistor NPN n, pilotato dal pin Px.y del microcontrollore.
COMMON	=	- Emitter comune dei transistors.
+VL	= I	- Tensione di alimentazione dei diodi di ricircolo (la stessa del carico).

Le linee di output a transistor, comprendono un diodo LED con funzione di feed-back visivo (il LED si accenderà tutte le volte in cui il transistor risulterà in conduzione); esse inoltre, sono optoisolate in modo da garantire una netta separazione galvanica tra l'elettronica interna ed il mondo esterno. Lo stadio finale di tali uscite é caratterizzato da un transistor Darlington **NPN** in Open Collector, dotato di diodo di ricircolo e con gli emettitori collegati ad un unico comune.

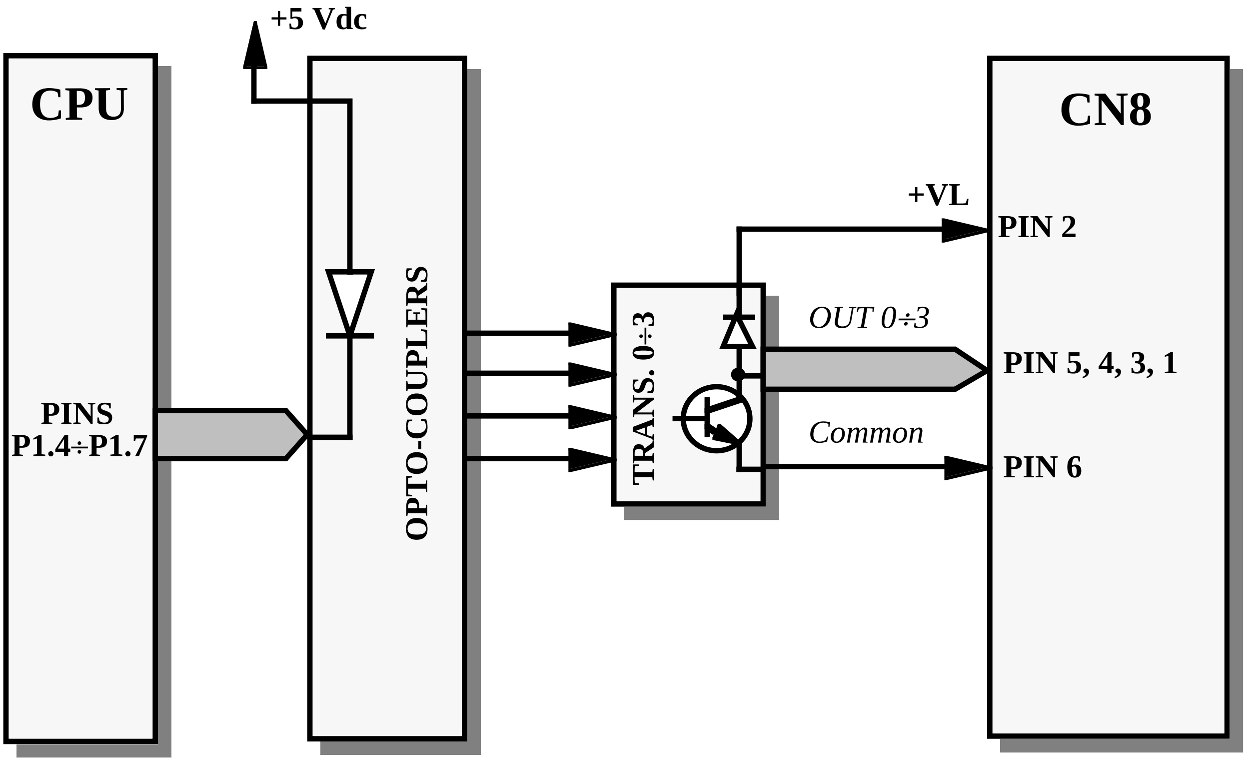


FIGURA 26: SCHEMA DELLE USCITE A TRANSISTORS

CN6 - CONNETTORE PER INGRESSO A/D CONVERTER E LINEE I/O TTL

CN6 é un connettore a morsettiera a rapida estrazione, composto da 2 contatti.
Tramite CN6 possono essere collegate le due linee multifunzione (ingresso di A/D converter, I/O TTL o ingressi del comporatore analogico) delle **GPC® R/T94**.
La circuiteria presente a bordo delle schede, é configurabile tramite i jumper J5 e J6 come descritto di seguito, in modo da risolvere le diverse problematiche dell'utente senza incrementi di costo.
Nel caso sia selezionato l'ingresso di conversione A/D, é da ricordare che questo é dotato di condensatore di filtro ed il segnale ad esso applicato deve essere compreso nel range 0÷10 V, 0÷20 mA o 4÷20 mA mentre la gestione della conversione viene realizzata acquisendo i registri del comparatore analogico del microprocessore come descritto nel capitolo "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO".

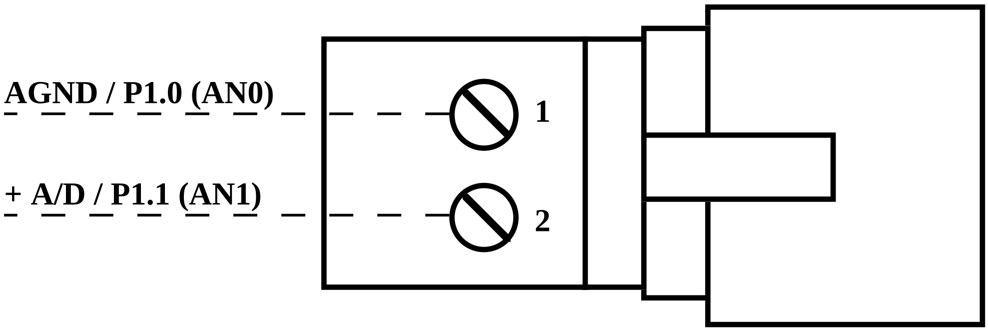


FIGURA 27: CN6- CONNETTORE PER INGRESSO A/D CONVERTER

Legenda:

Px.y (ANy) = I/O - Linea di I/O TTL collegata al pin Px.y del microcontrollore oppure ingresso ANy del comparatore analogico.
+ A/D = I - Linea di ingresso analogico della circuiteria di A/D converter.
AGND = - Linea di massa analogica.

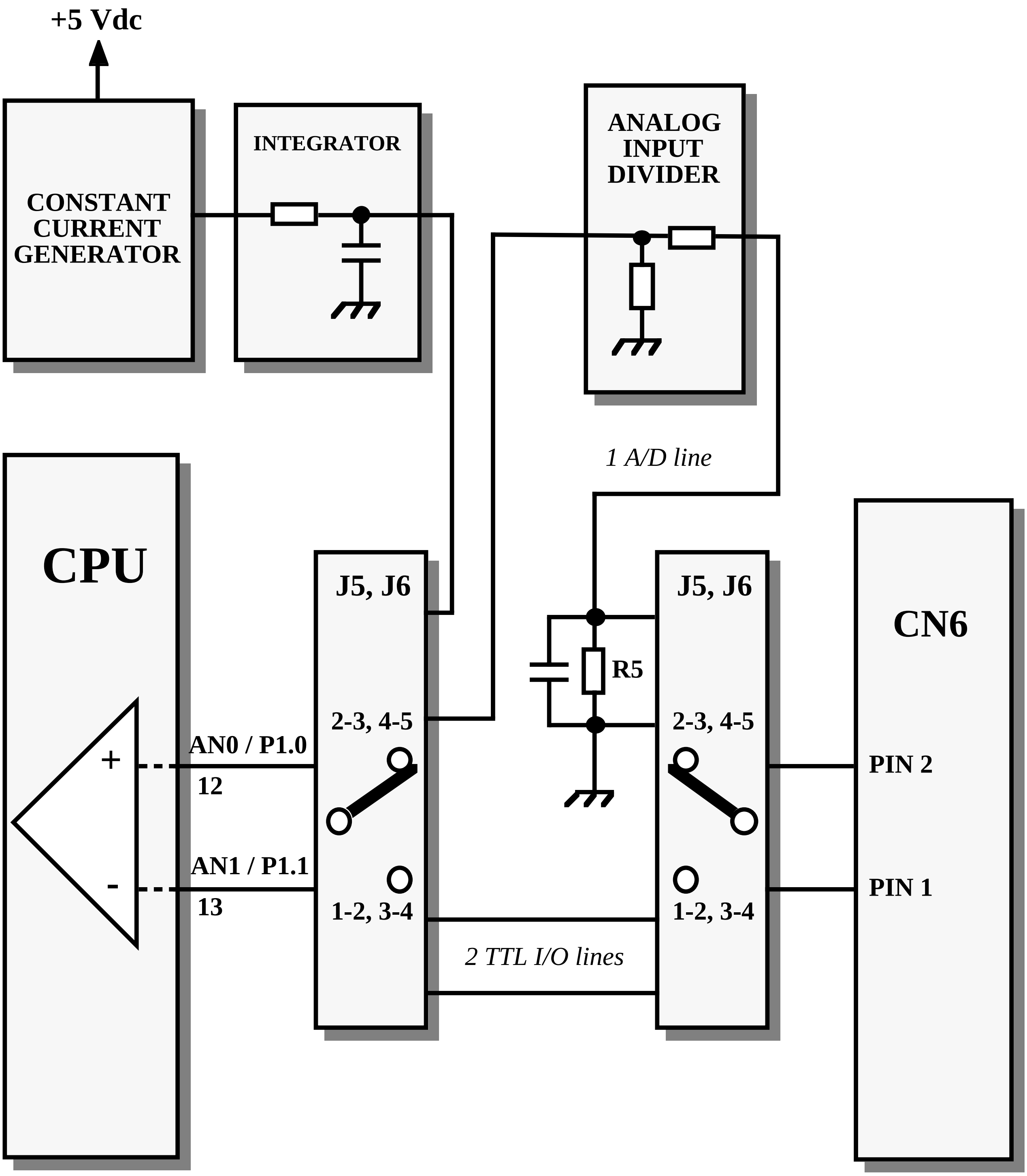


FIGURA 28: SCHEMA DELL'INGRESSO A/D CONVERTER

SEGNALAZIONI VISIVE

Le schede **GPC® R/T94** sono dotate di 13 LEDs con cui vengono segnalate alcune condizioni di stato, come descritto nella seguente tabella:

LEDs	COLORE	FUNZIONE
LD1÷LD4	Rosso	Segnalano lo stato delle linee di output digitali 0÷3. Il LED é attivo quando l'uscita OUTn é connessa a COMMON n.
LD5÷LD12	Verde	Segnalano lo stato delle linee di input digitali 0÷7. Il LED é attivo quando l'ingresso INPn é connesso a GND opto.
LD13	Verde	Segnala lo stato della linea di input AUX. Il LED é attivo quando l'ingresso AUX é connesso a GND opto.

FIGURA 29: TABELLA DELLE SEGNALAZIONI VISIVE

N.B.
I LEDs da 1 a 12 sono visibili sul profilo della scheda, mentre LD13 è visibile solamente togliendo il coperchio della scheda. La funzione principale di questi LEDs é quella di fornire un'indicazione visiva dello stato della scheda, facilitando quindi le operazioni di verifica di funzionamento di tutto il sistema. Per una più facile individuazione di tali segnalazioni visive a bordo scheda, si faccia riferimento alle figure 8 e 9.

INTERRUPTS

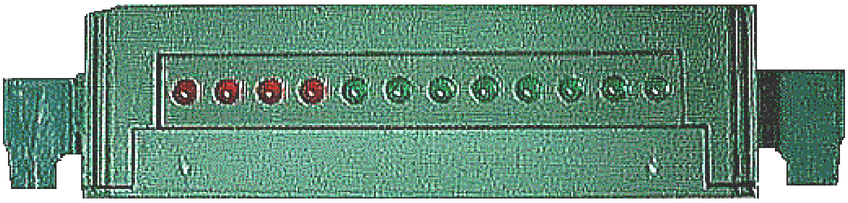
Una caratteristica peculiare delle **GPC® R/T94** è la notevole potenza nella gestione delle interruzioni. Di seguito viene riportata una breve descrizione di come possono essere gestiti i segnali hardware di interrupt della scheda; per quanto riguarda la gestione di tali interrupts si faccia riferimento ai data sheets del microprocessore oppure all'appendice B di questo manuale.

- Ingresso AUX -> Genera un interrupt sul pin /INT0 (P3.2) del microcontrollore.
- Periferiche della CPU -> Generano un interrupt interno. In particolare le possibili sorgenti d'interrupt interno sono le sezioni: Timer/Counter e linea seriale.

Sulla scheda é presente un gestore d'interrupt che consente di attivare, disattivare, mascherare le sorgenti d'interrupt e che regolamenta l'attivazione contemporanea di più interrupts. In questo modo l'utente ha sempre la possibilità di rispondere in maniera efficace e veloce a qualsiasi evento esterno, stabilendo anche la priorità delle varie sorgenti.

BACK UP

Le **GPC® R/T94** sono provviste di una batteria al litio BT1 che provvede a tamponare la SRAM+RTC di bordo anche in assenza della tensione di alimentazione. Il jumper J7 provvede a collegare o meno questa batteria in modo da salvaguardarne la durata prima dell'installazione o in tutti i casi in cui il back up non é necessario. Per l'indiduazione della batteria BT1 a bordo delle schede si vedano le figure 8 e 9.



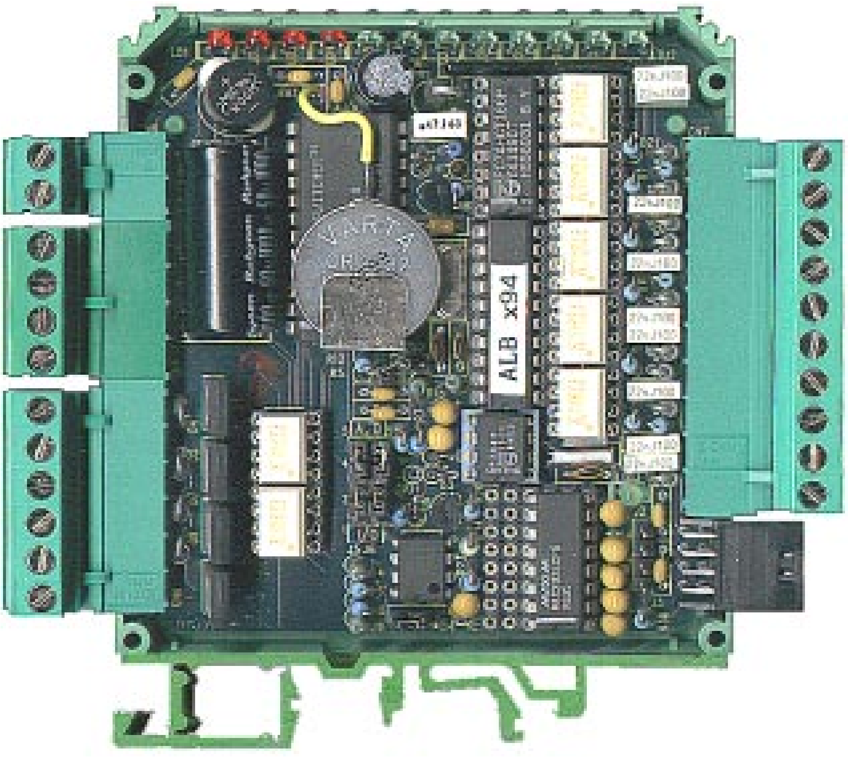


FIGURA 30: FOTO DELLA GPC® T94

INTERFACCIAMENTO DEGLI I/O CON IL CAMPO

Al fine di evitare eventuali problemi di collegamento della scheda con tutta l'elettronica del campo a cui le **GPC® R/T94** si devono interfacciare, si devono seguire le informazioni riportate nei precedenti paragrafi e le relative figure che illustrano le modalità interne di connessione.

- Per i segnali che riguardano la comunicazione seriale con i protocolli RS 232, RS 422, RS 485 e current loop, fare riferimento alle specifiche standard di ognuno di questi protocolli.

- Tutti i segnali a livello TTL possono essere collegati a linee dello stesso tipo riferite alla massa digitale della scheda. Il livello 0V corrisponde allo stato logico 0, mentre il livello 5V corrisponde allo stato logico 1.

- Il segnale d'ingresso alla sezione A/D devono essere collegati ad un segnale analogico che rispetti il range di variazione ammesso che può essere di 0÷10 V, 0÷20 mA o 4÷20mA a seconda della configurazione. Da notare tale ingresso presente su CN6, é dotato di condensatore di filtro che garantisce una maggiore stabilità sul segnale acquisito, ma che allo stesso tempo abbassa la frequenza di taglio.

- Per i segnali optoisolati d'ingresso, all'esterno devono essere collegati i soli contatti da acquisire. Tali contatti (relé, fine-corsa, interruttori, ecc.) devono collegare o meno il segnale d'ingresso INx alla GND opto.
  Per quanto rigurda la corrispondenza dei segnali logici, il contatto aperto genera un **1** logico, mentre il contatto chiuso genera uno **0** logico, secondo la normativa NPN.

- I segnali d'uscita a relé, presenti solo sulla **GPC® R94**, devono essere collegati direttamente al carico da pilotare (elettrovalvole, relé di potenza, teleruttori, ecc.). La scheda fornisce il contatto normalmente aperto, in grado di sopportare una corrente massima di **5A** con una tensione che può arrivare fino a **30 Vdc** oppure **250 Vac**.
  Per fornire la possibilità di pilotare anche carichi diversi, con alimentazioni distinte, sono previsti due diversi COMUNI relativi ad altrettanti gruppi di 2 relé.

- I segnali di uscita a transistor Darlington NPN, presenti solo sulla **GPC® T94**, devono essere collegati al carico da pilotare (elettrovalvole, relé di potenza, teleruttori, ecc.). La scheda fornisce la linea di output in Open Collector, in grado di sopportare una corrente massima di **4A non continuativi**, con una tensione che può arrivare fino a **+45 Vdc**.
  I transistors, essendo privi di radiatore, sono in grado di pilotare in maniera continuativa, un carico resistivo che, alimentato a **24 Vdc**, assorbe una corrente massima di **600 mA**, questo a condizione che la temperatura di lavoro rimanga a 20 gradi centigradi.

SELEZIONE TIPO INGRESSI ANALOGICI

Le schede **GPC® R/T94**, possono avere l'ingresso analogico in tensione o corrente, come descritto nei precedenti paragafi e capitoli. La selezione del tipo d'ingresso viene effettuata in fase di ordine della scheda montando su **R5**, una apposita resistenza di conversione tensione-corrente (codice opzione **.8420**); nel caso questo componente non sia montato (default) il corrispondente canale accetta un ingresso in tensione nei range 0÷10 V; viceversa un ingresso in corrente.
Il valore della resistenza di conversione corrente-tensione, si ottiene dalla seguente formula:

R = 10 V / Imax

Di norma il valore di questo componente é di **499 Ω**, relativo ad un ingresso di 4÷20 mA o 0÷20 mA.
Per eventuali esigenze al di fuori dei valori standard sopracitati si prega di contattare la **grifo®**.
Per l'individuazione della resistenza **R5** a bordo delle schede fare riferimento alle figure 8 e 9.

TENSIONI DI ALIMENTAZIONE

Le **GPC® R/T94** dispongono di una efficiente circuiteria che si presta a risolvere in modo comodo ed efficace il problema dell'alimentazione della scheda in qualsiasi condizione di utilizzo.
Di seguito vengono riportate le possibili configurazioni della sezione alimentatrice:

Senza alimentatore swicthing (default):

+V opto: Fornisce alimentazione agli optoisolatori della sezione di ingresso delle schede; deve essere di +24 Vdc e deve essere fornita tramite i pin 1 e 2 di CN4 oppure 2 e 8 di CN3.

+5 Vdc: Fornisce alimentazione alla logica di controllo ed alla sezione di output delle schede; deve essere di +5 Vdc ± 5% e deve essere fornita tramite i pin 1 e 7 di CN3.

Con alimentatore swicthing (opzione .SW):

+V opto: Fornisce alimentazione agli optoisolatori della sezione di ingresso delle schede; deve essere di +24 Vdc e deve essere fornita tramite i pin 1 e 2 di CN4 oppure 2 e 8 di CN3.

Vac: Fornisce alimentazione alla logica di controllo ed alla sezione di output delle schede, tramite l'alimentatore switching di bordo; deve essere di 10÷40 Vdc oppure 8÷24 Vac e deve essere fornita tramite i pin 3 e 4 di CN4 (in caso di tensione continua é indifferente la polarità). In questo modo é possibile alimentare le schede con dispositivi standard del settore industriale come trasformatori, batterie, celle solari, ecc. Se é necessario alimentare dei carichi esterni a +5 Vdc é possibile prelevare tale tensione dai pin 1 e 7 di CN3. Da notare che l'alimentatore switching di bordo é dotato di ponte radrizzatore a diodi, quindi in caso di alimentazione con una tensione continua, il segnale di massa digitale (GND) delle schede, non é allo stesso potenziale di quello presente su CN4.

Per garantire la massima immunità ai disturbi e quindi un corretto funzionamento delle schede, é necessario che queste due tensioni siano galvanicamente isolate tra di loro; a questo scopo può essere ordinato l'alimentatore **EXPS-2** che svolge questa funzione partendo dalla tensione di rete.

Le **GPC® R/T 94** sono dotate di una circuiteria di protezione a **TransZorb™** per evitare danni dovuti a tensioni non corrette. La selezione del tipo di sezione alimentatrice delle schede, deve avvenire in fase di ordine delle stesse; infatti questa scelta implica una diversa configurazione hardware che deve essere effettuata dal personale della **grifo®**.

JUMPERS

Esistono a bordo delle **GPC® R/T94** 8 jumpers, di cui 2 a stagno, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. Di seguito ne é riportato l'elenco, l'ubicazione e la loro funzione nelle varie modalità di connessione.

JUMPERS	N. VIE	UTILIZZO
J2, J3	2	Configurano il protocollo elettrico della linea seriale.
J4	3	Seleziona direzionalità e modalità di attivazione della linea seriale in RS 422, RS 485.
J5, J6	5	Configurano le 2 linee sul connettore CN6 come ingresso analogico A/D o I/O TTL generici.
J7	2	Collega la batteria al Litio di bordo alla circuiteria di back-up.
JS1, JS2	2	Collegano circuiteria di terminazione e forzatura RS 422, RS 485.

FIGURA 31: TABELLA RIASSUNTIVA JUMPERS

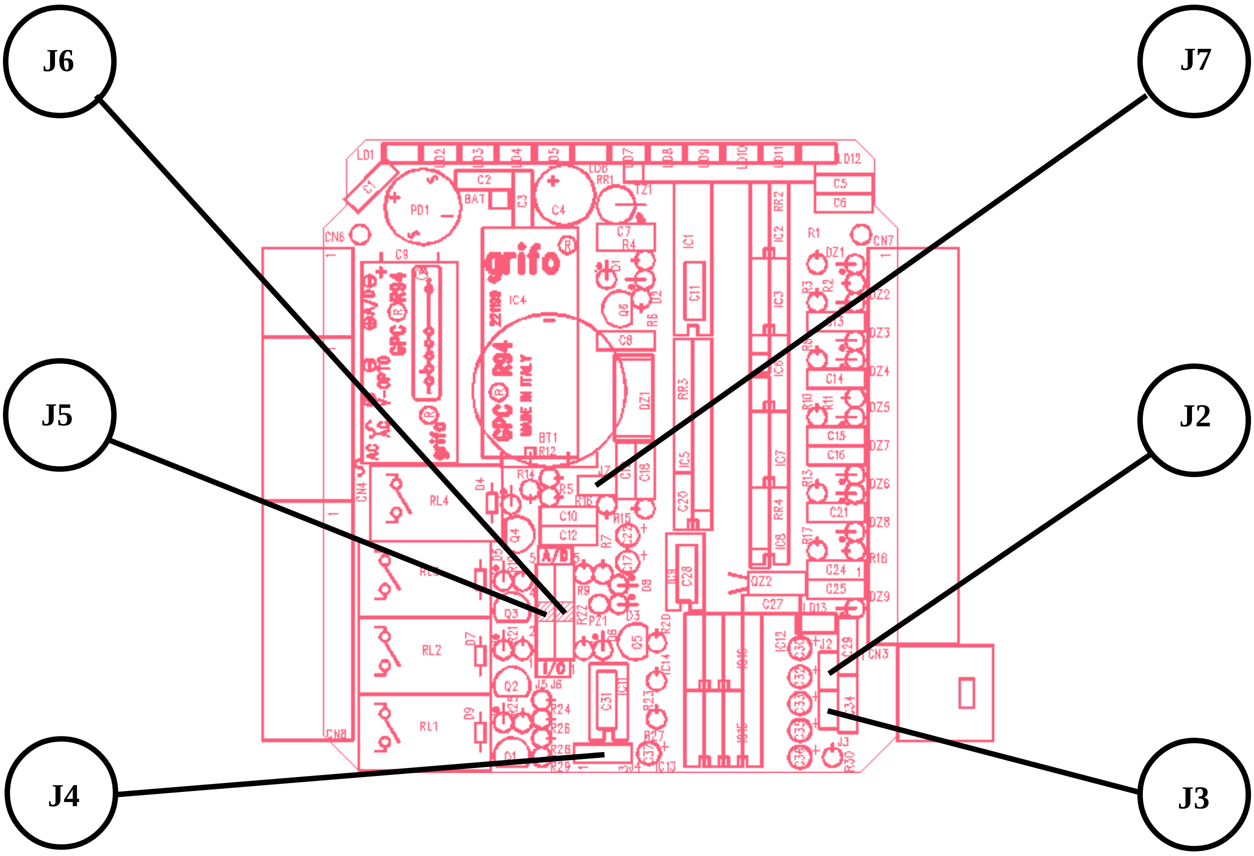
Di seguito é riportata una descrizione tabellare delle possibili connessioni degli 8 jumpers con la loro relativa funzione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa o alle figure 3÷6 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella seguente descrizione. Per l'individuazione dei jumpers a bordo della scheda, si utilizzino invece le figure 33 e 36. In tutte le seguenti tabelle l'***** indica la connessione di default, ovvero quella impostata in fase di collaudo, con cui la scheda viene fornita.

JUMPERS A 3 E 5 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J4	posizione 1-2	Seleziona la comunicazione sulla linea seriale in RS 485 (half duplex a 2 fili).	
	posizione 2-3	Seleziona la comunicazione sulla linea seriale in RS 422 (full duplex o half duplex a 4 fili).	*
J5, J6	posizioni 1-2, 3-4	Configurano i pin di CN6 come 2 linee di I/O TLL generici.	
	posizioni 2-3, 4-5	Configurano i pin di CN6 come ingresso analogico di conversione A/D.	*
	posizione 3-3	Collega, tra di loro, i pin P1.1 (AN1) e P1.0 (AN0) del microprocessore.	

FIGURA 32: TABELLA JUMPERS A 3 E 5 VIE

La connessione in posizione **3-3** descritta per i jumpers J5, J6 corrisponde ad unico ponticello a 2 vie che collega i pin 3 dei due jumpers in oggetto.



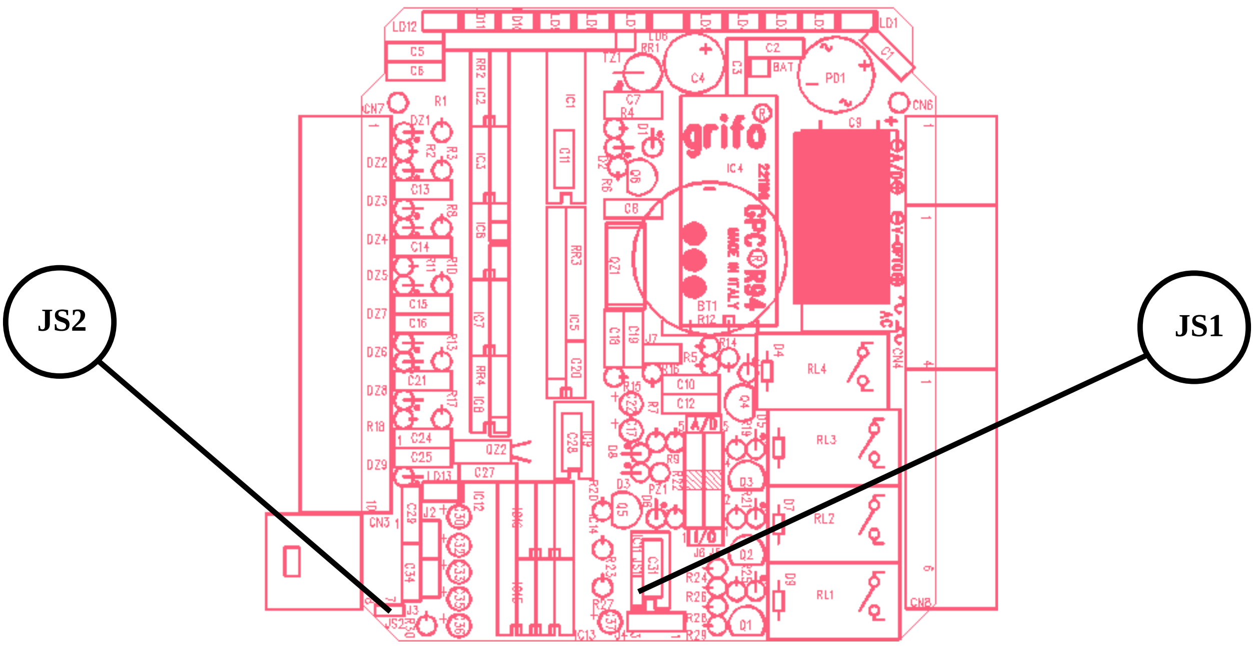


FIGURA 33: DISPOSIZIONE JUMPERS SU GPC® R94

JUMPERS A 2 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J2, J3	non connessi	Predispongono la linea seriale in RS232, RS422, RS485 o Current Loop (vedere il paragrafo "COMUNICAZIONE SERIALE" per maggiori informazioni).	*
	connessi	Predispongono la linea seriale in TTL (vedere il paragrafo"COMUNICAZIONE SERIALE" per maggiori informazioni).	
J7	non connesso	Non collega la batteria al litio BT1 alla circuiteria di back-up.	*
	connesso	Collega la batteria al litio BT1 alla circuiteria di back-up.	
JS1, JS2	non connessi	Non collegano la circuiteria di terminazione e forzatura alla linea di ricezione RS 422 o alla linea seriale RS 485.	*
	connessi	Collegano la circuiteria di terminazione e forzatura alla linea di ricezione RS 422 o alla linea seriale RS 485.	

FIGURA 34: TABELLA JUMPERS A 2 VIE

SELEZIONE MEMORIE

Le **GPC® R/T94** possono gestire fino ad un massimo di 5504 bytes di memoria variamente suddivisa. In particolare per la configurazione seguire le informazioni riportate nella seguente tabella:

IC	DISPOSITIVO	DIMENSIONE
5	AT89C2051	2K Bytes (FLASH EPROM) 128 bytes (SRAM)
	AT89C4051	4K Bytes (FLASH EPROM) 128 bytes (SRAM)
9	SRAM+RTC	256 Bytes
11	EEPROM	512÷1K Bytes

FIGURA 35: TABELLA DI SELEZIONE MEMORIE

Normalmente le **GPC® R/T94** sono fornita nella sua configurazione di default con il microprocessore AT89C4051, il dispositivo di SRAM+RTC montato e 512 bytes di EEPROM; ogni configurazione diversa può essere autonomamente variata dall'utente (ad esclusione della EEPROM IC11) oppure richiesta nella fase di ordine. Sotto sono riportate i codici delle opzioni di memoria disponibili:

.EE08 -> 1K EEPROM seriale

Per ulteriori informazioni e costi delle opzioni, contattare direttamente la **grifo®**, mentre per una facile individuazione dei dispositivi di memoria fare riferimento alle figure 8 e 9.

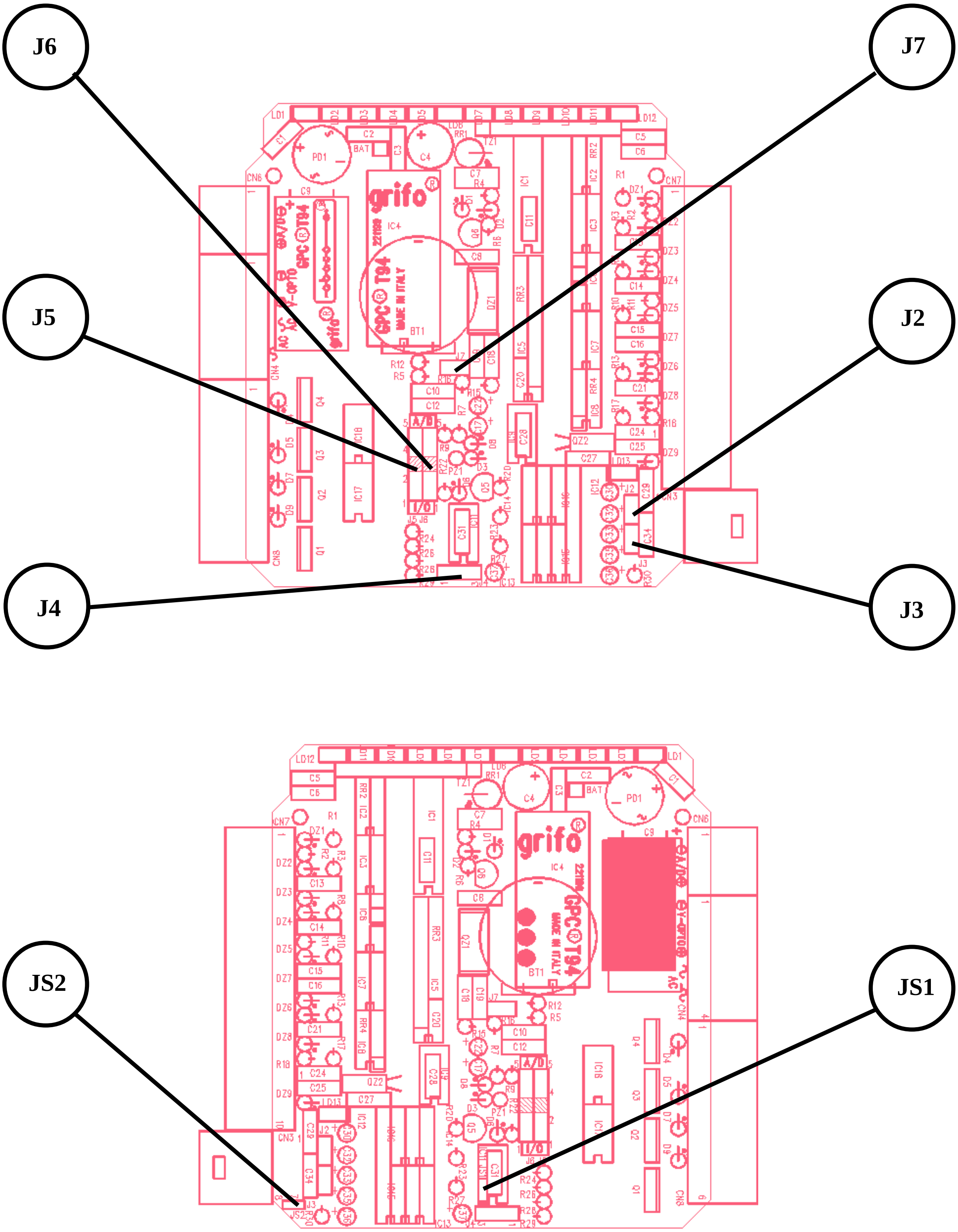


FIGURA 36: DISPOSIZIONE JUMPERS SU GPC® T94

SELEZIONE DEL TIPO DI COMUNICAZIONE SERIALE

La linea di comunicazione seriale delle schede **GPC® R/T94** può essere bufferata in TTL, in RS 232, in RS 422-485 o in current loop. La selezione del tipo d'interfacciamento avviene via hardware e viene effettuata tramite un opportuno strippaggio dei jumpers di bordo, come può essere desunto dalla lettura delle precedenti tabelle.
Dal punto di vista software sono invece definibili tutti i parametri del protocollo fisico di comunicazione tramite la programmazione dei registri interni del microprocessore.
Alcuni componenti necessari per le configurazioni RS 422, RS 485 e current loop non sono montati e collaudati sulla scheda in configurazione di default; per questo la prima configurazione della linea seriale non in RS 232 deve essere sempre effettuata dai tecnici **grifo®**. A questo punto l'utente può cambiare autonomamente la configurazione seguendo le informazioni sotto riportate:

- LINEA SERIALE SETTATA IN RS 232 (opzione **.RS232**)

			IC12	= driver MAX 202
J2, J3	=	non connessi	IC13	= nessun componente
J4	=	indifferente	IC14	= nessun componente
JS1, JS2	=	non connessi	IC15	= nessun componente
			IC16	= nessun componente

- LINEA SERIALE SETTATA IN CURRENT LOOP (opzione **.CLOOP**)

			IC12	= nessun componente
J2, J3	=	non connessi	IC13	= driver HP 4200
J4	=	indifferente	IC14	= driver HP 4100
JS1, JS2	=	non connessi	IC15	= nessun componente
			IC16	= nessun componente

Da ricordare che l'interfaccia seriale in current loop é di tipo passivo e si deve quindi collegare una linea current loop attiva, ovvero provvista di un proprio alimentatore. L'interfaccia current loop può essere utilizzata per realizzare sia connessioni punto punto che multipunto con un collegamento a 4 o 2 fili.

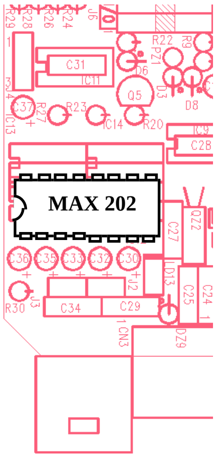
- LINEA SERIALE SETTATA IN RS 422 (opzione **.RS 422**)

			IC12	= nessun componente
J2, J3	=	non connessi	IC13	= nessun componente
J4	=	posizione 2-3	IC14	= nessun componente
JS1, JS2	=	(*1)	IC15	= driver MAX 483 o SN 75176
			IC16	= driver MAX 483 o SN 75176

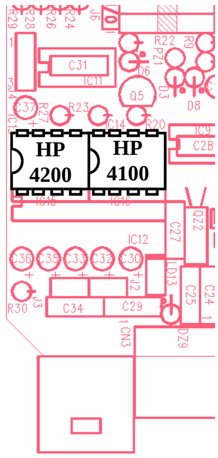
Lo stato del pin P3.7 del microprocessore, gestito via software, consente di abilitare o disabilitare il trasmettitore come segue:

P3.7	= livello basso	= stato logico 0	->	trasmettitore attivo
P3.7	= livello alto	= stato logico 1	->	trasmettitore disattivo

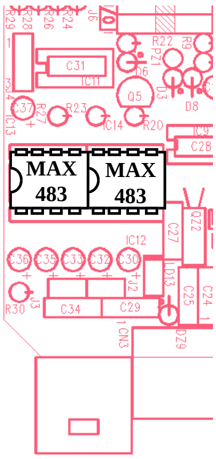
Per sistemi punto punto, la linea P3.7 può essere mantenuta sempre bassa (trasmettitore sempre attivo), mentre per sistemi multipunto si deve attivare il trasmettitore solo in corrispondenza della trasmissione.



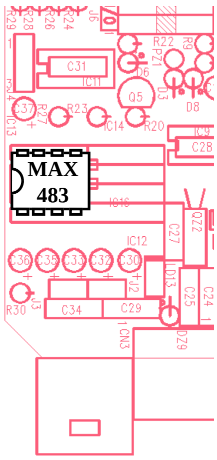
Seriale in RS 232



Seriale in Current-Loop



Seriale in RS 422



Seriale in RS 485

FIGURA 37: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE

- LINEA SERIALE SETTATA IN RS 485 (opzione **.RS 485**)

			IC12	= nessun componente
J2, J3	=	non connessi	IC13	= nessun componente
J4	=	posizione 1-2	IC14	= nessun componente
JS1, JS2	=	(*1)	IC15	= driver MAX 483 o SN 75176
			IC16	= nessun componente

In questa modalità le linee da utilizzare sono i pin 5 e 6 di CN3, che quindi diventano le linee di trasmissione o ricezione a seconda dello stato del pin P3.7 del microprocessore, gestito via software, come segue:

P3.7	= livello basso	= stato logico 0	->	linea in trasmissione
P3.7	= livello alto	= stato logico 1	->	linea in ricezione

Questa comunicazione la si utilizza sia per connessioni punto punto che multipunto con un collegamento a 2 fili. Sempre in questa modalità é possibile ricevere quanto trasmesso, in modo da fornire al sistema la possibilità di verificare autonomamente la riuscita della trasmissione; infatti in caso di conflitti sulla linea, quanto trasmesso non viene ricevuto correttamente e viceversa.

(*1) Nel caso si utilizzi la linea seriale in RS 422 o RS 485, con i jumpers JS1 e JS2 é possibile connettere la circuiteria di terminazione e forzatura sulla linea . Tale circuiteria deve essere sempre presente nel caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione.

- LINEA SERIALE SETTATA TTL (configurazione default)

			IC12	= nessun componente
J2, J3	=	connessi	IC13	= nessun componente
J4	=	indifferente	IC14	= nessun componente
JS1, JS2	=	non connessi	IC15	= nessun componente

In fase di power on, il segnale P3.7 é mantenuto a livello logico alto di conseguenza in seguito ad questa fase il driver RS 485 é in ricezione oppure il driver di trasmissione RS 422 é disattivo, in modo da eliminare eventuali conflittualità sulle linee di comunicazione.
Per ulteriori informazioni relative alla comunicazione seriale fare riferimento agli esempi di collegamento delle figure 12÷19 ed alla appendice B di questo manuale.

DESCRIZIONE SOFTWARE

Le schede **GPC® R/T94** hanno la possibilità di usufruire di una ricca serie di strutture software che consentono di utilizzarne al meglio le caratteristiche. In generale la scheda può sfruttare tutte le risorse software per il microprocessore montato e tutti i pacchetti ideati per la famiglia 51, sia ad alto che a basso livello. Tra questi ricordiamo:

MICRO/ASM-51: Macro Cross Assembler. Disponibile in ambiente MS-DOS e nella versione "ASSOLUTA" o "RILOCABILE", permette una facile ed efficiente programmazione in assembler, dei microcontrollori basati sull'8051. In versione "RILOCABILE", viene anche fornito un LINKER ed un GESTORE DI LIBRERIE.

MICRO/C-51: Integer Cross Compiler per files sorgenti scritti in linguaggio "C". Disponibile in ambiente MS-DOS, genera un source assembly compatibile con il MICRO/ASM-51 o con il Macro Assembler rilocabile dell'Intel (MCS-51).

MICRO/SLD-51: Simulatore e Debugger a livello source. Simulatore/Debugger in grado di simulare i microcontrolloridella famiglia I51 e di monitorare lo stato di esecuzione di un programma. Permette tramite un PC e senza l'aggiunta di emulatori o hardware addizionale, il caricamento o il salvataggio di file HEX o SIMBOLICI, il settaggio di breakpoints, l'esecuzione in modalità TRACE di istruzioni "C" e/o "ASSEMBLER", la visualizzazione di qualsiasi registro o variabile, ecc.

HI-TECH C: Cross compilatore per file sorgenti scritti in linguaggio "C". E' un potente pacchetto software che tramite un comodo I.D.E. permette di utilizzare un editor, un compilatore "C" (floating-point), un assemblatore, un linker e un remote debugger. Sono inoltre inclusi i source delle librerie.

BASCOM-8051: Cross compilatore a basso costo per files sorgenti scritti in BASIC,disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore ed un simulatore molto potente per il debugger del source.

DDS C: E' un comodo pacchetto software, a basso costo, che tramite un completo I.D.E. permette di utilizzare un editor, un compilatore "C" (integer), un assemblatore, un linker e un remote debugger abbinato ad un monitor. Sono inclusi i sorgenti delle librerie ed una serie di utility.

DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO

Di seguito viene riportata una descrizione dettagliata, relativa alla gestione software delle varie periferiche di bordo delle **GPC® R/T94**. Qualora la documentazione riportata fosse insufficiente fare riferimento direttamente alla documentazione tecnica della casa costruttrice del componente. In questo paragrafo inoltre non vengono descritte le sezioni che fanno parte del microprocessore; per quanto riguarda la programmazione di quest'ultime si faccia riferimento all'appendice B di questo manuale.

EEPROM SERIALE

Per quanto riguarda la gestione del modulo di EEPROM seriale (IC11), si faccia riferimento alla documentazione specifica del componente. In questo manuale tecnico non viene riportata alcuna informazione software in quanto la modalità di gestione è articolata e prevede una conoscenza approfondita del componente e comunque l'utente può usare le apposite procedure ad alto livello fornite nei pacchetti di programmazione. Si ricorda solo che i primi 32 bytes (0÷31) sono riservati e perciò si deve evitare la modifica dei medesimi. La logica di controllo consente la gestione software della EEPROM tramite due pin di I/O del microprocessore, con le seguenti corrispondenze:

P1.3	<->	linea DATA	(SDA)
P1.2	->	linea CLOCK	(SCL)

Data l'implementazione hardware della circuiteria di gestione del modulo di EEPROM seriale, si ricorda che di tale dispositivo i segnali **A0**,**A1** dello slave address sono posti a **0** logico, mentre il segnale **A2** é posto ad 1 logico. Lo stato logico 0 dei bit corrisponde allo stato logico basso (=0 V) del relativo segnale, mentre lo stato logico 1 corrisponde allo stato logico alto (=5 V) del segnale. I pin di I/O sono ad 1 logico in fase di power on, di conseguenza in seguito a questa fase i due segnali dell'EEPROM sono allo stato logico alto.

SRAM+RTC SERIALE

Per quanto riguarda la gestione del modulo di SRAM+RTC seriale (IC 9), si faccia riferimento alla documentazione specifica del componente. In questo manuale tecnico non viene riportata alcuna informazione software in quanto la modalità di gestione è articolata e prevede una conoscenza approfondita del componente e comunque l'utente può usare le apposite procedure ad alto livello fornite nei pacchetti di programmazione. La logica di controllo consente la gestione software della EEPROM tramite due pin di I/O del microprocessore, con le seguenti corrispondenze:

P1.3	<->	linea DATA	(SDA)
P1.2	->	linea CLOCK	(SCL)

Data l'implementazione hardware della circuiteria di gestione del modulo di SRAM+RTC, si ricorda che di tale dispositivo il segnale **A0** dello slave address é posto a **0** logico. Lo stato logico 0 dei bit corrisponde allo stato logico basso (=0 V) del relativo segnale, mentre lo stato logico 1 corrisponde allo stato logico alto (=5 V) del segnale. I pin di I/O sono ad 1 logico in fase di power on, di conseguenza in seguito a questa fase i due segnali della SRAM+RTC sono allo stato logico alto.

INGRESSI DIGITALI

Lo stato dei 9 ingressi digitali NPN, viene acquisito tramite il driver serializzatote 74 HCT 166 (ingressi IN0÷7) o direttamente da un pin di I/O del microcontrollore (ingresso AUX), con le seguenti corrispondenze:

Linea **H** 74 HCT 166	<-	Ingresso IN0
Linea **G** 74 HCT 166	<-	Ingresso IN1
Linea **F** 74 HCT 166	<-	Ingresso IN2
Linea **E** 74 HCT 166	<-	Ingresso IN3
Linea **D** 74 HCT 166	<-	Ingresso IN4
Linea **C** 74 HCT 166	<-	Ingresso IN5
Linea **B** 74 HCT 166	<-	Ingresso IN6
Linea **A** 74 HCT 166	<-	Ingresso IN7
Pin **P3.2** (**/INT0**) CPU	<-	Ingresso AUX

Quando gli input NPN sono attivi (contatto d’ingresso chiuso verso GND opto), le corrispondenti linee si trovano allo stato logico basso (0 logico), viveversa quando gli input sono disattivo (contatto d’ingresso aperto), viene acquisito un livello alto (1 logico).
Come detto in precedenza i LEDs LD6÷12 forniscono un indicazione visiva dello stato degli ingressi digitali (LED acceso=ingresso attivo).
Per quanto riguarda la gestione del driver serializzatore 74 HCT 166, si faccia riferimento alla documentazione specifica del componente. In questo manuale tecnico non viene riportata alcuna informazione software in quanto la modalità di gestione è articolata e prevede una conoscenza approfondita del componente e comunque l'utente può usare le apposite procedure ad alto livello fornite nei pacchetti di programmazione. La logica di controllo consente la gestione software di questo componente tramite tre pin di I/O del microprocessore, con le seguenti corrispondenze:

P3.3	<-	Linea DATA
P3.4	->	Linea LOAD
P3.5	->	Linea CLK

Lo stato logico 0 dei bit corrisponde allo stato logico basso (=0 V) del relativo segnale, mentre lo stato logico 1 corrisponde allo stato logico alto (=5 V) del segnale. I pin di I/O sono ad 1 logico in fase di power on, quindi in seguito a questa fase i segnali del 74 HCT 166 sono allo stato logico alto.

A titolo di esempio viene riportata di seguito una procedura, scritta per il compilatore **BASCOM 8051**, che restituisce lo stato degli ingressi digitali IN0÷7 gestendo il 74 HCT 166.

```
' Dichiarazione delle costanti
Dt_166 Alias P3.3          ' Linea DATA del 74 HCT 166
Ld_166 Alias P3.4          ' Linea LOAD del 74 HCT 166
Clk_166 Alias P3.5         ' Linea CLK del 74 HCT 166

' Dichiarazione delle variabili
Dim In0_7 As Byte          ' Stato degli ingressi IN0..IN7
Dim I As Byte              ' Contatore generico
Dim Flg As Bit             ' Flag di uso generico
```

```
' Dichiarazione delle procedure
Declare Sub Init_166
Declare Sub Inp_166


' Procedura: INIT_166
' Inizializza le linee di gestione del 74 HCT 166
' Input:  --
' Output: --

Sub Init_166
  Clk_166 = 1              ' Setta il Clock a 1
  Ld_166 = 1               ' Linea Load disattiva
  Dt_166 = 1               ' Setta la linea Data in input
  Flg = Dt_166
Return


' Procedura: INP_166
' Restituisce lo stato degli ingressi IN0..IN7
' Input:  --
' Output: In0_7 = Stato di IN0 (bit 0)..IN7 (bit7)

Sub Inp_166
  In0_7 = 0                ' Azzera la variabile
  Clk_166 = 0              ' Stato di partenza del Clock
  Ld_166 = 0               ' Deve acquisisce il byte in ingresso
  Clk_166 = 1              ' Fronte del Clock per acquisire il dato
  Ld_166 = 1
  For I = 1 To 8           ' Loop per acquisire gli 8 bit
    Flg = Dt_166           ' Legge un bit
    Clk_166 = 0            ' Pone il Clock a 0
    If Flg = 1 Then        ' Setta il bit In0_7.7 con lo stato di Flg
      Set In0_7.7
    Else
      Reset In0_7.7
    End If
    If I <> 8 Then
      clr c
      Rotate In0_7 , Right , 1        ' Shifta il byte a destra di 1
      Clk_166 = 1                      ' Fronte del Clock per prossima lettura
    End If
  Next I
Return
```

USCITE DIGITALI

Lo stato delle 4 uscite digitali, a relé o transistors NPN, viene definito tramite la gestione di altrettanti pins di I/O del micorcontrollore, con le seguenti corrispondenze:

P1.4	->	Uscita OUT0	(relé RL1 o transistor Q1)
P1.5	->	Uscita OUT1	(relé RL2 o transistor Q2)
P1.6	->	Uscita OUT2	(relé RL3 o transistor Q3)
P1.7	->	Uscita OUT3	(relé RL4 o transistor Q4)

Quando le linee di I/O vengono settati allo stato logico basso (0 logico), l'uscita corrispondenteviene attivata (transistors in conduzione o contatto del relé connesso al relativo comune), viceversa quando i pins si trovano allo stato logico alto (1 logico) le uscite OUTn sono disattive (transistors non in coduzione o contatto del relé aperto).
Come detto in precedenza i LEDs LD1÷4 forniscono un indicazione visiva dello stato degli output digitali (LED acceso=uscita attiva).
I pin di I/O sono ad 1 logico in fase di power on, di conseguenza in seguito a questa fase i 4 relé o transistors si trovano disattivati.

INGRESSO DI CONFIGURAZIONE

Connettendo i jumper J5 e J6 in posizione 3-3, come descritto nel precedente capitolo, vengono cortocircuitati tra di loro i pin **P1.0** (AN0) e **P1.1** (AN1) del microprocessore; quindi gestendo opportunamente questa condizione é possibile distinguere due diverse modalità di funzionamento. Ovviamente questo é possibile solo se una delle due modalità (quella attivata dalle linee cortocircuitare), non usa nessuna delle funzioni disponibili su questi segnali (A/D converter, comparatore analogico o I/O TTL). Ad esempio nel firmware **ALB x94**, questa caratteristica é utilizzata per distinguere la modalità RUN da quella di SETUP.
Da un punto di vista software, essendo abbastanza articolata la procedura di acquisizione dell'ingresso di configurazione si consiglia, in caso di necessità, di contattare direttamente la **grifo®**.

PERIFERICHE DELLA CPU

La descrizione dei registri e del relativo significato di tutte le periferiche interne della CPU (Timer Counter, controllore interrupts, linea seriale, ports di I/O, ecc) é disponibile nell'appendice B. Qualora queste informazioni fossero ancora insufficienti, fare riferimento alla documentazione tecnica della casa costruttrice.

A/D CONVERTER

La gestione dell'A/D converter, viene effettuata tramite il comparatore analogico ed un timer di bordo del microcontrollore. Per quanto riguarda la gestione dei relativi registri si faccia riferimento all'appendice B di questo manuale. Mentre per le modalità di conversione e la relativa gestione software, in questo manuale tecnico non viene riportata alcuna informazione in quanto é prevista una conoscenza approfondita, anche a livello teorico, della problematica. Tali argomentazioni sono state abbondantemente trattate dall'ATMEL in application notes, disponibili su Internet o nei Data-Book e comunque l'utente può usare le apposite procedure ad alto livello fornite nei pacchetti di programmazione.

A titolo di esempio viene riportata di seguito una procedura, scritta per il compilatore **BASCOM 8051**, che restituisce la combinazione digitale corrispondente al segnale analogico in ingresso. In questo caso é utilizzato il timer 0 per generare la base dei tempi necessaria.

```
' Dichiarazione delle costanti
Tr0 Alias Tcon.4            ' Flag start/stop del TIMER 0
Tf0 Alias Tcon.5            ' Flag Over-flow del TIMER 0
An0 Alias P1.0              ' Ingresso capacita'di carica dell'A/D
An1 Alias P1.1              ' Ingresso analogico dell'A/D

' Dichiarazione delle procedure
Declare Sub AD_Conv


' Procedura: AD_CONV
' Restituisce la conversione acquisita dall'A/D in un tempo massimo di 60 msec
' Input:  --
' Output: TH0, TL0 = Combinazione acquisita

Sub AD_Conv
  Tr0 = 0                   ' Ferma il Timer 0
  Tf0 = 0                   ' Azzera il flag di Over-flow
  Th0 = 0                   ' Azzerati i registri del Timer 0
  Tl0 = 0
  An1 = 1                   ' Sblocca l'ingresso analogico dell'A/D
  An0 = 1                   ' Fa partire la carica del condensatore
  Tr0 = 1                   ' Fa partire il Timer 0
                            ' Inizio della routine assembler
!Ad_2:
  jb tf0 , ad_3  ;          ' Verifica fine conversione da Over-flow
  jb p3.6 , ad_3 ;          ' Verifica fine conversione testando se il
                            ' bit P3.6 (uscita comparatore) e' = 1
  Sjmp ad_2
!Ad_3:
                            ' Fine della routine assembler
  Tr0 = 0                   ' Ferma il Timer 0
```

```
  If Tf0 = 1 Then          ' Se il Timer 0 e' in Over-flow, setta....
    Th0 = 255              ' ....la conversione a fondo scala
    Tl0 = 255
  End If
  Tf0 = 0                  ' Azzera il flag di Over-flow
  An0 = 0                  ' Scarico il condensatore per circa 6 msec
  Waitms 6
End Sub
```

Il firmware **ALB x94** (**ABACO® L**ink **B**US), permette di sfruttare tutte le risorse della scheda tramite una serie di comandi sulla linea seriale; per questo motivo tutto quanto ricevuto viene interpretato ed eseguito, ed eventuali codici di risposta vengono restituiti all'unità master di controllo.
E' inoltre prevista, una modalità di funzionamento SETUP, che permette di configurare, nel modo desiderato, tutte le varie sezioni del dispositivo.

SELEZIONE DELLA MODALITA' DI FUNZIONAMENTO

Il firmware **ALB x94** é in grado di gestire due modalità distinte, in particolare può essere attivato il modo **SETUP** di configurazione, oppure il modo **RUN** di normale funzionamento. La selezione dell'uno o dell'altro funzionamento, avviene durante la fase di **Power-ON**, verificando lo stato logico in cui si trovano i jumper **J5** e **J6**.
La selezione della modalità di funzionamento avviene come di seguito descritto:

Modo SETUP:	*J5 e J6 in posizione 2-3 e 4-5* *Pin 3-3 di J5 e J6 CONNESSI*
Modo RUN:	*J5 e J6 in posizione indifferente* *Pin 3-3 di J5 e J6 NON CONNESSI*

MODALITA' DI FUNZIONAMENTO SETUP

Tramite il modo **SETUP** é possibile definire tutti i vari parametri d' inizializzazione, relativi al settaggio del Baud Rate, del Tipo di comunicazione e del Nome del dispositivo. Tali dati saranno memorizzati all'interno della EEPROM, e permetteranno di definire la configurazione di lavoro nel modo RUN. Per una corretta programmazione dei vari parametri seguire i seguenti punti:

1) Togliere alimentazione alla **GPC® R/T 94**.

2) Assicurarsi che tutti gli ingressi e uscite della scheda non siano connessi ad alcun segnale.

3) Selezionare la modalità di funzionamento SETUP con i jumper J5 e J6, come descritto nel precedente paragrafo.

4) Fornire alimentazione alla **GPC® R/T 94**.

5) Se la modalità SETUP é stata riconosciuta viene attivata l' uscita OUT0 (LD1 acceso).
A questo punto l'utente deve impostare il BAUD RATE ed il TIPO DI COMUNICAZIONE settando gli ingressi IN0÷IN7 seguendo le indicazioni riportate nelle seguenti tabelle.

N.B.
Il settaggio desiderato deve rimanere impostato fino al punto successivo.

BAUD RATE	IN0	IN1	IN2	IN3	IN4	IN5
38400 Baud	OFF	OFF	OFF	OFF	OFF	OFF
19200 Baud	ON	OFF	OFF	OFF	OFF	OFF
9600 Baud	OFF	ON	OFF	OFF	OFF	OFF
4800 Baud	OFF	OFF	ON	OFF	OFF	OFF
2400 Baud	OFF	OFF	OFF	ON	OFF	OFF
1200 Baud	OFF	OFF	OFF	OFF	ON	OFF

FIGURA 38: TABELLA DI SELEZIONE DEL BAUD RATE

TIPO DI COMUNICAZIONE	IN6	IN7
Punto-Punto	OFF	OFF
Master-Slave 9 bits	ON	OFF

FIGURA 39: TABELLA DI SELEZIONE DEL TIPO DI COMUNICAZIONE

Da notare che le indicazioni ON ed OFF indicano rispettivamente gli ingressi connessi e non connessi verso la GND opto.
Per confermare e salvare i parametri selezionati, l'utente deve chiudere l'ingresso AUX verso GND opto e successivamente riaprirlo.

3) Se l' operazione precedente non ha avuto successo viene attivato anche OUT3 (LD4 acceso) e l'utente deve ripetere il punto 2. Al contrario se l'operazione ha dato esito positivo risulta attiva l'uscita OUT1 (LD2 acceso).
A questo punto é necessario immettre il NOME utilizzato nella comunicazione Master-Slave; per l'inserimento di questo paramertro (valori consenti128÷255) si deve utilizzare la stessa tecnica del punto 2, ossia si devono settare opportunamente le linee di input e si deve confermare il parametro con un impulso sull' ingresso AUX.
Da notare che il NOME viene immesso seguendo una codifica binaria (l'ingresso INx in posizione **ON** corrisponde al relativo bit a **0**), come riportato nella seguente tabella e che il valore impostato non é significativo nel caso sia stata precedentemente selezionata la comunicazione punto-punto.

NOME	IN0	IN1	IN2	IN3	IN4	IN5	IN6	IN7
128	ON	ON	ON	ON	ON	ON	ON	OFF
129	OFF	ON	ON	ON	ON	ON	ON	OFF
...	...	...	...	...	...	...	...	...
255	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

FIGURA 40: TABELLE SETTAGGIO DEL NOME

4) Se l' operazione precedente non ha avuto successo viene attivato anche OUT3 (LD4 acceso) e l'utente deve ripetere il punto 3. Al contrario se l'operazione ha dato esito positivo vengono salvati tutti i parametri nella EEPROM di bordo della scheda.
Se questa operazione non é riuscita vengono attivate tutte le uscite OUT0÷OUT3 (LD1÷LD4 accesi) al contrario viene settata la seguente condizione:

OUT0	=	Attiva	(LD1 acceso)
OUT1	=	Disattiva	(LD2 spento)
OUT2	=	Attiva	(LD3 acceso)
OUT3	=	Disattiva	(LD4 spento)

5) Togliere l' alimentazione alla scheda.

6) Impostare la modalità di funzionamento RUN per il normale utilizzo della scheda.

MODALITA' DI FUNZIONAMENTO RUN

Quando si accede al modo **RUN**, vengono verificati i parametri memorizzati nell'EEPROM, relativi al protocollo di comunicazione. Se questi non sono validi (EEPROM non inizializzata) la scheda entra in un loop di attesa; a questo punto é possibile solo togliere alimentazione dal dispositivo.

NOTA BENE

Di default l'EEPROM non é inizializzata, perciò come prima operazione, l'utente deve provvedere configurare correttamente la scheda (modalità SETUP).

Il protocollo di comunicazione del modo RUN é settabile in termini di Baud Rate e tipo di comunicazione (mediante la modalità SETUP), mentre il formato dei dati é in funzione della modalità selezionata, in particolare:

Comunicazione Punto-Punto:	**8 bit, 1 Stop, NO Parity**
Master-Slave:	**9 bit, 1 Stop, NO Parity**

Nella modalità RUN, le periferiche della scheda vengono gestite con le modalità illustate di seguito.

Linea seriale:
E' utilizzata per ricevere i comandi e trasmettere le eventuali risposte; é possibile comunicare in modalità punto-punto oppure master-slave. Quest'ultima modalità consente di realizzare una rete di telecontrollo i cui singoli moduli, remotati anche a notevole disatanza,sono pilotatati da un'unica unità master (PC, PLC, scheda della serie **GPC®**, ecc.).

Uscite digitali OUT0÷OUT3:
Possono venire settate contemporaneamente con un unico comando, oppure é possibile una gestione delle singole linee. In questo caso sono disponibili anche una serie di comandi che permettono di realizzare delle attivazioini temporizzate, delle onde quadre ecc.

Ingressi digitali IN0÷IN7:
Vengono acquisiti contemporaneamente con un unico comando, oppure sono disponibili dei comandi per la gestione a bit dei singoli ingressi.

Ingresso digitale AUX:
Può essere utilizzato come linea di ingresso digitale, oppure come segnale di clock per un contatore a 16 bit, gestito da appositi comandi.

EEPROM seriale:
Sono disponbili una serie di comandi per permettono di gestire i bytes dell'EEPROM in termini di messaggi da 10 caratteri. Questo permette di memorizzare in memoria non volatile, informazioni come stringhe di testo, settaggi, parametri, ecc.

SRAM+RTC seriale:
Sono disponibili una serie di comandi che permettono di gestire il Real Time Clock in termini di ore, minuti, secondi, giorno, mese, anno e giorno della settimana. Inoltre é possibile leggere i scrivere i bytes della SRAM,di cui questo dispositivo é dotato.

Linee di I/O TTL:
Sono utizzate per l'implementazione di due distinti BUS per la comunicazione tramite il protocollo **1-Wire**®. Sono quindi disponibili unaserie di comandi che permettono di gestire i vari dispositivi collegabili a tali BUS (sensori di temperatura, memorie, Dallas iButton™, ecc.).

I comandi riconosciuti nella modalità di funzionamento RUN, sono descritti nei seguenti paragrafi.

COMANDI PER FUNZIONI VARIE

Sono di seguito riportati i comandi per la gestione di varie funzione.

RESET GENERALE

Sequenza del comando:

Codice Dec:	**65**	**97**
Codice Hex:	**41**	**61**
Mnemonico:	**A**	**a**

A seguito di questo comando, il firmware ripristina la condizione di inizializzazione che si verifica all'atto del Power-ON; in paricolare:

OUT0÷OUT3: Vengono disattivati e posti nella condizione iniziale, quindi settati allo stato logico 0. Eventuali temporizzazioni vengono interrotte.

Counter a 16 Bit: Viene inizializzato con il valore 0.

RICHIESTA DISPONIBILITA' A SCRIVERE IN EEPROM

Sequenza del comando:

Codice Dec:	**66**
Codice Hex:	**42**
Mnemonico:	**B**

Mediante questo comando é possibile richiedere alla scheda se é pronta per scrivere delle informazioni nella EEPROM di bordo; esso deve essere eseguito prima di inviare uno dei comandi che comportano una scrittura in questo tipo di dispositivo.

Codici di risposta:
I possibili codici restituito a seguito di tale comando sono i seguenti:

0 *EEPROM non disponibile per gestire un nuovo messaggio*
1 *EEPROM disponibile per gestire un nuovo messaggio*

SCRITTURA DEL BYTE DI PRESENZA

Sequenza del comando:

Codice Dec:	**89**	**dat.L**	**dat.H**
Codice Hex:	**59**	**dat.L**	**dat.H**
Mnemonico:	**Y**	**ASCII(dat.L)**	**ASCII(dat.H)**

Il byte di presenza della scheda viene settato con il valore indicato nel parametro "dat" che può essere compreso nel range **0÷255**.
Questo byte é un'allocazione riservata nella EEPROM di bordo che, una volta settata con il valore desiderato, permette, ad esempio, di verificare che la **GPC® R/T94** funziona correttamente, oppure che non ci siano conflitti di comunicazione sulla linea seriale.
Da notare il parametro "dat" deve essere inviato a nibble, in particolare viene trasmesso prima il nibble basso (**dat.L** = bit 0÷3) poi quello alto (**dat.H** = bit 4÷7).

N.B.
Tale comando comporta la scrittura di un dato nella EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova scrittura su tale dispositivo; se così non é, il comando viene ignorato.

Esempio:
Se si desidera memorizzare, il dato 65 sarà necessario inviare la seguente sequenza: **89 1 4**.

LETTURA DEL BYTE DI PRESENZA

Sequenza del comando:

Codice Dec:	**121**
Codice Hex:	**79**
Mnemonico:	**y**

La **GPC® R/T94** restituisce in seriale il valore del proprio byte di presenza.
Tale comando può essere utile, ad esempio, nel caso si debba verificare la presenza, o il corretto funzionamento della scheda stessa.

Codici di risposta:
Viene restituito il byte, compreso nel range 0÷255, tramite una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**dat.L** = bit 0÷3) poi quello alto (**dat.H** = bit 4÷7).

LETTURA DELLA VERSIONE DEL FIRMWARE

Sequenza del comando:

Codice Dec:	**86**
Codice Hex:	**56**
Mnemonico:	**V**

Codici di risposta:
Vengono restituiti i due codici **Y** e **X** che compongono il numero di versione **X.Y** del firmware.

LETTURA DEL CODICE SCHEDA

Sequenza del comando:

Codice Dec:	**75**
Codice Hex:	**4B**
Mnemonico:	**K**

Codici di risposta:
Viene restituito il valore **1** che é il codice identificativo delle schede **GPC® R/T94**.

SETTAGGIO CONFIGURAZIONE DEL PIN AUX

Sequenza del comando:

Codice Dec:	**85**	**cfg**
Codice Hex:	**55**	**cfg**
Mnemonico:	**U**	**ASCII(cfg)**

Configura l'ingresso digitale AUX delle **GPC® R/T 94**, come linea di input oppure come segnale di clock per il conter interno a 16 bits. Tale settaggio é indicato dal parametro "cfg", in particolare:

0	*Pin AUX settato come linea di INPUT*
2	*Pin AUX settato come CLOCK per il counter a 16 Bits*

Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

N.B.
Tale comando comporta la scrittura di un dato nella EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova scrittura su tale dispositivo; se così non é, il comando viene ignorato.

Esempio:
Se si vuole settare la linea **AUX** come clock é necessario inviare la seguente sequenza: **85 2**.

LETTURA CONFIGURAZIONE DEL PIN AUX

Sequenza del comando:

Codice Dec:	**102**
Codice Hex:	**66**
Mnemonico:	**f**

Restitusce la configurazione attuale dell'ingresso digitale AUX delle **GPC® R/T 94**.

Codici di risposta:
Il byte restituito può assumere i seguenti valori:

0	*Pin AUX attualmente settato come linea di INPUT*
2	*Pin AUX attualmente settato come CLOCK per il Counter a 16 Bits*

COMANDI PER GESTIONE DEI PORT DI I/O

Sono riportati di seguito i comandi relativi alla gestione dei port di I/O digitale delle **GPC® R/T94**.

SCRITTURA DEL PORT DI OUTPUT

Sequenza del comando:

Codice Dec:	**87**	**1**	**dat**	**0**
Codice Hex:	**57**	**1**	**dat**	**0**
Mnemonico:	**W**	**SOH**	**ASCII(dat)**	**NUL**

Viene scritto il port di output con il byte "dat"; secondo la seguente corrispondenza:

(MSB) 0 0 0 0 **OUT3 OUT2 OUT1 OUT0** (LSB)

Dove **OUTn** indica lo stato logico, **0** (output disattivo) o **1** (output attivo), che devono assumere le relative uscite a relé o transistors delle schede.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Per attivare le uscite OUT0 e OUT3 sarà necessario inviare la seguente sequenza: **87 1 9 0**.

LETTURA DEL PORT DI INPUT

Sequenza del comando:

Codice Dec:	**82**	**0**
Codice Hex:	**52**	**0**
Mnemonico:	**R**	**NUL**

Viene acquisito e quindi restituito lo stato degli ingressi digitali NPN IN0÷7 delle **GPC® R/T94**.

Codici di risposta:
Il dato acquisito dal port di input viene restituito sotto forma di due nibble secondo con il seguente formato:

Nibble L:	(MSB)	0	0	0	0	**IN3**	**IN2**	**IN1**	**IN0**	(LSB)
Nibble H:		0	0	0	0	**IN7**	**IN6**	**IN5**	**IN4**	

Dove i bit **INn**, indicano gli stati logici **0** (ON = ingresso chiuso verso GND opto) o **1** (OFF = ingresso aperto), in cui si trovano le relative linee di input NPN.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole leggere il port di input, sul quale é presente il dato **90** (5A Hex) sarà necessario inviare la seguente sequenza: **82 0**, e come risposta si avranno i seguenti bytes: **10 5**.

COMANDI PER GESTIONE A BIT DEI PORT DI I/O

Sono riportati di seguito i comandi relativi alla gestione a bit delle singole linee dei port di I/O digitale delle **GPC® R/T94**.

SET DI UN BIT DEL PORT DI OUTPUT

Sequenza del comando:

Codice Dec:	**83**	**1**	**bit**
Codice Hex:	**53**	**1**	**bit**
Mnemonico:	**S**	**SOH**	**ASCII(bit)**

Viene posta allo stato logico **1** (uscita attiva), la linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
Se sulla linea di output in questione sono in corso delle temporizzazioni, queste vengono interrotte.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole attivare la linea **OUT2** é necessario inviare la seguente sequenza: **83 1 2**.

SET TEMPORIZZATO DI UN BIT DEL PORT DI OUTPUT

Sequenza del comando:

Codice Dec:	**115**	**1**	**bit**	**tmp.L**	**tmp.H**
Codice Hex:	**73**	**1**	**bit**	**tmp.L**	**tmp.H**
Mnemonico:	**s**	**SOH**	**ASCII(bit)**	**ASCII(tmp.L)**	**ASCII(tmp.H)**

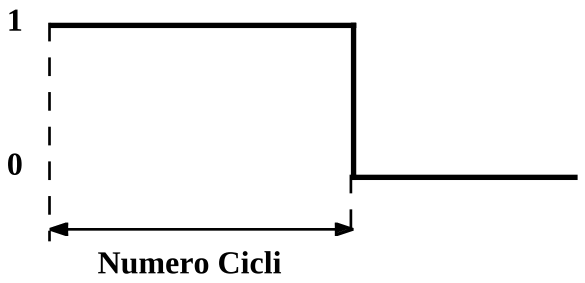


FIGURA 41: COMANDO DI SET TEMPORIZZATO

Viene posta allo stato logico **1** (uscita attiva), per un determinato tempo, la linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
L'uscita selezionata rimane in questo stato per il numero di cicli (1 ciclo = **10 msec**) determinato dal byte "tmp", quindi ritorna allo stato logico **0** (uscita disattiva).
Il valore di questa temporizzazione deve essere compreso nel range **1÷255** e deve essere inviato a <u>nibble</u>, in particolare prima il nibble basso (**tmp.L** = bit 0÷3) poi quello alto (**tmp.H** = bit 4÷7).
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole attivare la linea **OUT2** per un tempo di 500 msec, corrispondente a **50** cicli é necessario inviare la seguente sequenza: **115 1 2 2 3**.

CLEAR DI UN BIT DEL PORT DI OUTPUT

Sequenza del comando:

Codice Dec:	**67**	**1**	**bit**
Codice Hex:	**43**	**1**	**bit**
Mnemonico:	**C**	**SOH**	**ASCII(bit)**

Viene posta allo stato logico **0** (uscita disattiva), la linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
Se sulla linea di output in questione sono in corso delle temporizzazioni, queste vengono interrotte.
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole disattivare la linea **OUT2** é necessario inviare la seguente sequenza: **67 1 2**.

CLEAR TEMPORIZZATO DI UN BIT DEL PORT DI OUTPUT

Sequenza del comando:

Codice Dec:	**99**	**1**	**bit**	**tmp.L**	**tmp.H**
Codice Hex:	**63**	**1**	**bit**	**tmp.L**	**tmp.H**
Mnemonico:	**c**	**SOH**	**ASCII(bit)**	**ASCII(tmp.L)**	**ASCII(tmp.H)**

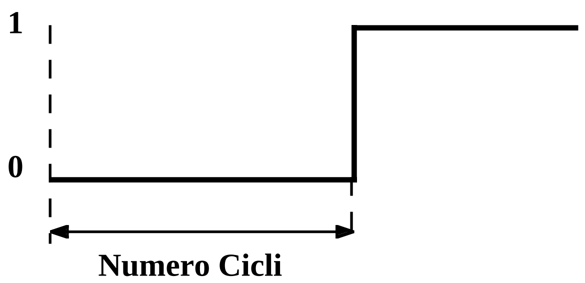


FIGURA 42: COMANDO DI CLEAR TEMPORIZZATO

Viene posta allo stato logico **0** (uscita disattiva), per un determinato tempo, la linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
L'uscita selezionata rimane in questo stato per il numero di cicli (1 ciclo = **10 msec**) determinato dal byte "tmp", quindi ritorna allo stato logico **1** (uscita attiva).
Il valore di questa temporizzazione deve essere compreso nel range **1÷255** e deve essere inviato a nibble, in particolare prima il nibble basso (**tmp.L** = bit 0÷3) poi quello alto (**tmp.H** = bit 4÷7).
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole disattivare la linea **OUT2** per un tempo di 500 msec, corrispondente a **50** cicli é necessario inviare la seguente sequenza: **99 1 2 2 3**.

LETTURA DI UN BIT DEL PORT DI INPUT

Sequenza del comando:

Codice Dec:	**114**	**port**	**bit**
Codice Hex:	**72**	**port**	**bit**
Mnemonico:	**r**	**ASCII(port)**	**ASCII(bit)**

Viene acquisito e restituito lo stato dell'ingresso digitale NPN indicato dai parametri "port" e "bit", con la corrispondenza:

Input IN0 ÷ IN7	->	***port** = 0*	***bit** = 0 ÷ 7*
Input AUX	->	***port** = 3*	***bit** = 0*

Se la sequenza inviata contiene dei dati al diversi da quelli indicati, il comando viene ignorato.

Codici di riposta:
Viene restituito lo stato logico dell'input indicato; in particolare viene restuito il valore **0** (ingresso chiuso verso GND opto) oppure **1** (ingresso aperto).

Esempio:
Se si vuole leggere lo stato della linea AUX é necessario inviare la seguente sequenza: **114 3 0**.

LETTURA CON DEBOUCING DI UN BIT DEL PORT DI INPUT

Sequenza del comando:

Codice Dec:	**68**	**port**	**bit**	**tmp.L**	**tmp.H**
Codice Hex:	**44**	**port**	**bit**	**tmp.L**	**tmp.H**
Mnemonico:	**D**	**ASCII(port)**	**ASCII(bit)**	**ASCII(tmp.L)**	**ASCII(tmp.H)**

Viene acquisito e restituito, lo stato dell'ingresso digitale NPN indicato dai parametri "port" e "bit", con la corrispondenza:

Input IN0 ÷ IN7	->	***port** = 0*	***bit** = 0 ÷ 7*
Input AUX	->	***port** = 3*	***bit** = 0*

A differenza del comando precedente l'acquisizione viene eseguita con un processo di deboucing, la cui durata, espressa in cicli (1 ciclo=**10 msec**), é determinata dal parametro "tmp".
Tale byte deve essere compreso nel range **1÷255** e deve essere inviato a nibble, in particolare prima il nibble basso (**tmp.L** = bit 0÷3) poi quello alto (**tmp.H** = bit 4÷7).
Se la sequenza inviata contiene dei dati al diversi da quelli indicati, il comando viene ignorato.

Codici di riposta:
Il dato che viene restituito può assumere i seguenti valori:

0	*Ingresso chiuso verso GND opto durante tutto il tempo di deboucing*
1	*Ingresso aperto durante tutto il tempo di deboucing*
7	*Lo stato dell'ingresso é variato durante il tempo di deboucing*

Esempio:
Se si vuole leggere lo stato della linea IN2, con un tempo di deboucing di 50 msec, corrispondente a 5 cicli, é necessario inviare la seguente sequenza: **68 0 2 5 0**.

ONDA QUADRA TEMPORIZZATA DA 1 SU BIT DEL PORT OUTPUT

Sequenza del comando:

Codice Dec:	**112**	**1**	**bit**	**tmp.L**	**tmp.H**
			sta.L	**sta.H**	
Codice Hex:	**70**	**1**	**bit**	**tmp.L**	**tmp.H**
			sta.L	**sta.H**	
Mnemonico:	**p**	**SOH**	**ASCII(bit)**	**ASCII(tmp.L)**	**ASCII(tmp.H)**
			ASCII(sta.L)	**ASCII(sta.H)**	

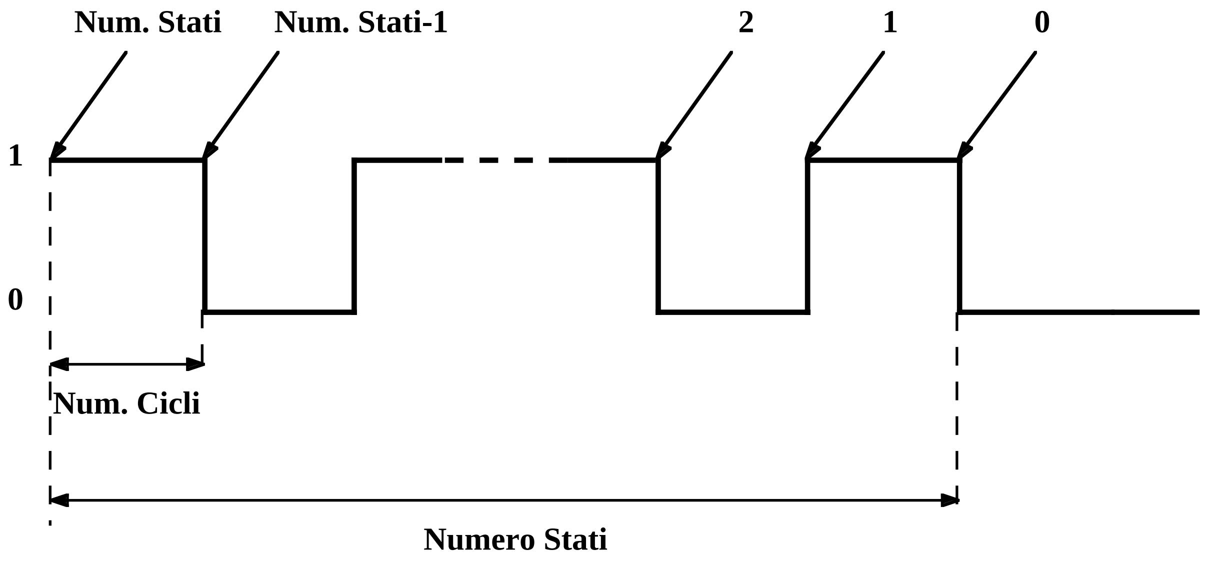


FIGURA 43: COMANDO DI ONDA QUADRA TEMPORIZZATA DA 1

Viene generata un'onda quadra, **con stato logico iniziale 1** e duty-cicle del 50% (uscita attiva e disattiva alternativamente ad intervalli uguali), sulla linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
Il numero di cicli del semi-periodo (1 ciclo = **10 msec**) viene indicato nel byte "tmp".
Il valore di questa temporizzazione deve essere compreso nel range **1÷255** e deve essere inviato a nibble, in particolare prima il nibble basso (**tmp.L** = bit 0÷3) poi quello alto (**tmp.H** = bit 4÷7).
il parametro "sta" invece indica il numero di cambiamenti di stato che devono avvenire sull'uscita indicata; facendo riferimento alla figura si può notare che vengono eseguite **"sta"+1** variazioni.
Anche questo dato deve essere compreso nel range **1÷255** e deve essere inviato a nibble, in particolare prima il nibble basso (**sta.L** = bit 0÷3) poi quello alto (**sta.H** = bit 4÷7).
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole attivare questo comando sulla linea **OUT2** per un tempo di 200 msec, corrispondente a **20** cicli e con **10** cambiamenti di stato, é necessario inviare la sequenza: **112 1 2 4 1 9 0**.

N.B.
Facendo riferimento alla figura si può notare che lo stato che assumerà l'uscita al termine dell'esecuzione del comando dipende dal numero di variazioni eseguite; in particolare un numero pari di cambiamenti comporterà uno stato finale **0** (output disattivo) e viceversa.

ONDA QUADRA TEMPORIZZATA DA 0 SU BIT DEL PORT OUTPUT

Sequenza del comando:

Codice Dec:	**119**	**1**	**bit**	**tmp.L**	**tmp.H**
			sta.L	**sta.H**	
Codice Hex:	**77**	**1**	**bit**	**tmp.L**	**tmp.H**
			sta.L	**sta.H**	
Mnemonico:	**w**	**SOH**	**ASCII(bit)**	**ASCII(tmp.L)**	**ASCII(tmp.H)**
			ASCII(sta.L)	**ASCII(sta.H)**	

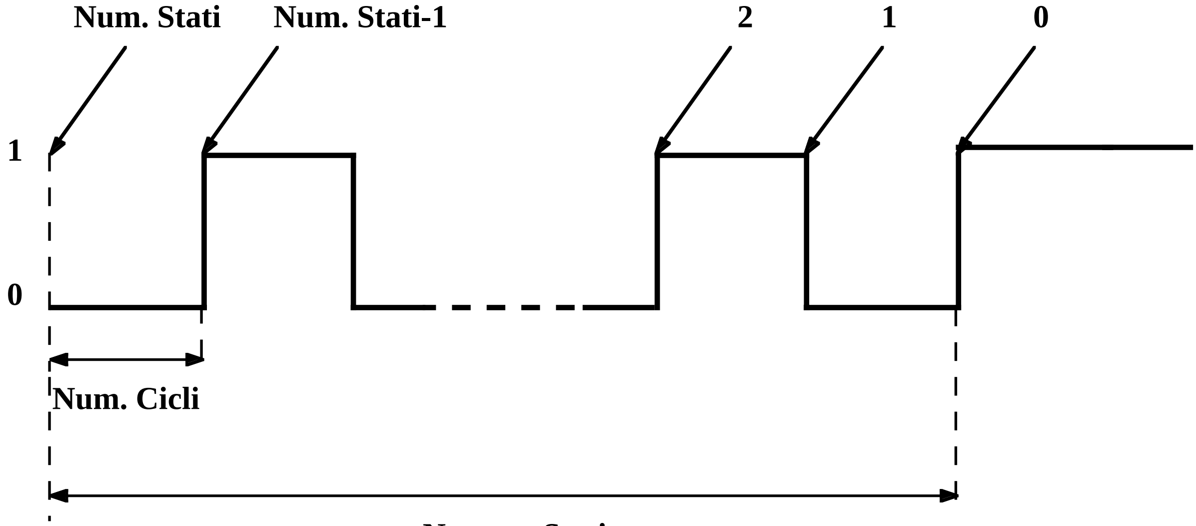


FIGURA 44: COMANDO DI ONDA QUADRA TEMPORIZZATA DA 0

Viene generata un’onda quadra, **con stato logico iniziale 0** e duty-cicle del 50% (uscita attiva e disattiva alternativamente ad intervalli uguali), sulla linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
Il numero di cicli del semi-periodo (1 ciclo = **10 msec**) viene indicato nel byte "tmp".
Il valore di questa temporizzazione deve essere compreso nel range **1÷255** e deve essere inviato a <u>nibble</u>, in particolare prima il nibble basso (**tmp.L** = bit 0÷3) poi quello alto (**tmp.H** = bit 4÷7).
il parametro "sta" invece indica il numero di cambiamenti di stato che devono avvenire sull’uscita indicata; facendo riferimento alla figura si può notare che vengono eseguite **"sta"+1** variazioni.
Anche questo dato deve essere compreso nel range **1÷255** e deve essere inviato a <u>nibble</u>, in particolare prima il nibble basso (**sta.L** = bit 0÷3) poi quello alto (**sta.H** = bit 4÷7).
Se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole attivare questo comando sulla linea **OUT2** per un tempo di 200 msec, corrispondente a **20** cicli e con **10** cambiamenti di stato, é necessario inviare la sequenza: **119 1 2 4 1 9 0**.

N.B.
Facendo riferimento alla figura si può notare che lo stato che assumerà l’uscita al termine dell’esecuzione del comando dipende dal numero di variazioni eseguite; in particolare un numero pari di cambiamenti comporterà uno stato finale **1** (output attivo) e viceversa.

ONDA QUADRA SU UN BIT DEL PORT DI OUTPUT

Sequenza del comando:

Codice Dec:	**80**	**1**	**bit**	**tmp.L**	**tmp.H**
Codice Hex:	**50**	**1**	**bit**	**tmp.L**	**tmp.H**
Mnemonico:	**P**	**SOH**	**ASCII(bit)**	**ASCII(tmp.L)**	**ASCII(tmp.H)**

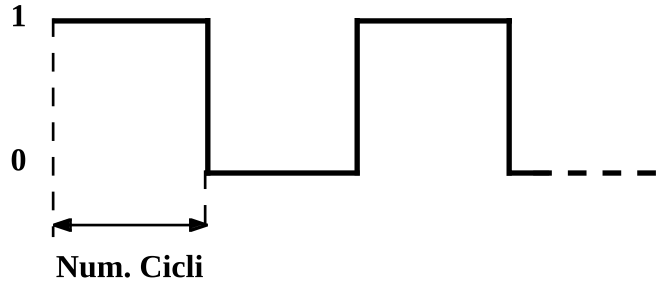


FIGURA 45: COMANDO DI ONDA QUADRA

Viene generata un'onda quadra con duty-cicle del 50% (uscita attiva e disattiva alternativamentead intervalli uguali), sulla linea di output OUTn, dove n é il numero indicato in "bit"; tale parametro può assumere i valori **0**, **1**, **2** e **3**.
Il numero di cicli del semi-periodo (1 ciclo = **10 msec**) viene indicato nel byte "tmp".
Il valore di questa temporizzazione deve essere compreso nel range **1÷255** e deve essere inviato a nibble, in particolare prima il nibble basso (**tmp.L** = bit 0÷3) poi quello alto (**tmp.H** = bit 4÷7).

Esempio:
Se si vuole attivare questo comando sulla linea **OUT2** per un tempo di 200 msec, corrispondente a **20** cicli é necessario inviare la seguente sequenza: **80 1 2 4 1**.

COMANDI PER GESTIONE DEL COUNTER A 16 BITS

Sono di seguito riportati i comandi relativi alla gestione del counter a 16 bits.
Tale contatore viene incrementato dalle variazioni di stato sul pin AUX, quando questo é configurato come segnale di clock.

LETTURA DEL CONTATORE A 16 BITS

Sequenza del comando:

Codice Dec:	**73**
Codice Hex:	**49**
Mnemonico:	**I**

Restituisce il valore attuale del contatore a 16 bits.

Codici di risposta:
La sequenza restituita a seguito del comando, è formata da 4 codici che esprimono la combinazione a 16 bit attualmente memorizzata all'interno dei registri del contatore; questa é ricevuta a nibble secondo il seguente formato:

Contatore	(bit 0÷3)	:	(MSB)	0	0	0	0	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	(LSB)
	(bit 4÷7)	:		0	0	0	0	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	
	(bit 8÷11)	:		0	0	0	0	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**	
	(bit 12÷15)	:		0	0	0	0	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	

Quando il contatore raggiunge il suo massimo valore, corrispondente a 65535 (FFFF Hex), ad un successivo impulso sul clock, il nuovo valore della combinazione sarà 0.
Se il pin AUX é configurato come input, a seguito di questo comando, sarà sempre restituita la combinazione 0.

Esempio:
Se i registri del contatore a 16 bit contegono la combinazione 23055 (5A0F Hex), a seguito della trasmissione del comando **73**, sarà restituita la seguente sequenza: **15 0 10 5**.

RESET DEL CONTATORE A 16 BITS

Sequenza del comando:

Codice Dec:	**88**	**120**
Codice Hex:	**58**	**78**
Mnemonico:	**X**	**x**

A seguito di questo comando, il vengono azzerati i registri del counter a 16 bit, scrivendo in essi la combinazione **0**.

COMANDI PER LA GESTIONE DEI MESSAGGI

Di seguito sono riportati i comandi relativi alla gestione dei messaggi da 10 caratteri nella EEPROM seriale delle **GPC® R/T94**.

LETTURA NUMERO ULTIMO MESSAGGIO MEMORIZZABILE

Sequenza del comando:

Codice Dec:	**77**
Codice Hex:	**4D**
Mnemonico:	**M**

Questo comando permette di acquisire il numero dell'ultimo messaggio memorizzabile nella EEPROM. Questo varia in funzione del dispositivo installato e corrisponde a quanto riportato nella seguente tabella.

EEPROM	N.MAX
24c02 (256 Bytes)	23
24c04 (512 Bytes)	48
24c08 (1024 Bytes)	99

FIGURA 46: NUMERO DELL'ULTIMO MESSAGGIO MEMORIZZABILE

Codici di riposta:
Il numero dell'ultimo messaggio memorizzabile viene restituito con una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**carn.L** = bit 0÷3) poi quello alto (**carn.H** = bit 4÷7).

MEMORIZZAZIONE DI UN MESSAGGIO

Sequenza del comando:

Codice Dec:	**69**	**msg.L**	**msg.H**			
		car0.L	**car0.H**	**...**	**car9.L**	**car9.H**
Codice Hex:	**45**	**msg.L**	**msg.H**			
		car0.L	**car0.H**	**...**	**car9.L**	**car9.H**
Mnemonico:	**E**	**ASCII(msg.L)**	**ASCII(msg.H)**			
		ASCII(car0.L)	**ASCII(car0.H)**	**...**	**ASCII(car9.L)**	**ASCII(car9.H)**

Il messaggio, di 10 caratteri, il cui numero è indicato in "msg" viene memorizzato nella EEPROM. Il valore di questoparametro deve essere compreso nel range **0÷N.MAX** e deve essere inviato a nibble, in particolare prima il nibble basso (**msg.L** = bit 0÷3) poi quello alto (**msg.H** = bit 4÷7). Con "N.MAX" si intende il numero dell'ultimo messaggio memorizzabile, visto in precedenza, e ricavabile direttamente dalla tabella precedente o tramite l'apposito comando.

I codici dei dieci caratteri che compongono il messaggio possono essere compresi in tutto il range **0÷255** e devono essere inviati a nibble, in particolare prima il nibble basso (**carn.L** = bit 0÷3) poi quello alto (**carn.H** = bit 4÷7).
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

N.B.
Tale comando comporta la scrittura di un dato nella EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova scrittura su tale dispositivo; se così non é, il comando viene ignorato.

Esempio:
Se si desidera memorizzare, come messaggio numero 16, la stringa "ABCDEFGHIJ" (corrispondente ai codici: 65, 66, 67, 68, 69, 70, 71, 72, 73, 74), sarà necessario inviare la seguente sequenza:
69 0 1 1 4 2 4 3 4 4 4 5 4 6 4 7 4 8 4 9 4 10 4.

LETTURA DI UN MESSAGGIO

Sequenza del comando:

Codice Dec:	**76**	**msg.L**	**msg.H**
Codice Hex:	**4C**	**msg.L**	**msg.H**
Mnemonico:	**L**	**ASCII(msg.L)**	**ASCII(msg.H)**

Il messaggio, di 10 caratteri, il cui numero è indicato in "msg" viene letto dalla EEPROM di bordo e quindi restituito.
Il valore di questo parametro deve essere compreso nel range **0÷N.MAX** e deve essere inviato a nibble, in particolare prima il nibble basso (**msg.L** = bit 0÷3) poi quello alto (**msg.H** = bit 4÷7).
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Codici di risposta:
I codici dei 10 caratteri che formano il messaggio vengono restituiti con una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**carn.L** = bit 0÷3) poi quello alto (**carn.H** = bit 4÷7).

Esempio:
Facendo riferimento all'esempio del comando precedente, se si vuole leggere il messaggio numero **16** dalla EEPROM, sarà necessario inviare la seguente sequenza: **76 0 1**. La risposta a seguito del tale comando sarà la sequenza: **1 4 2 4 3 4 4 4 5 4 6 4 7 4 8 4 9 4 10 4**.

COMANDI PER LA GESTIONE DEL SRAM+RTC

Sono riportati di seguito i comandi relativi alla gestione del modulo seriale di SRRAM+RTC delle schede **GPC® R/T94**.

SETTAGGIO OROLOGIO

Sequenza del comando:

Codice Dec:	**79**	**ore.L**	**ore.H**	**min.L**	**min.H**
		sec.L	**sec.H**	**gio.L**	**gio.H**
		mes.L	**mes.H**	**ann.L**	**ann.H**
		gse.L	**gse.H**		
Codice Hex:	**4F**	**ore.L**	**ore.H**	**min.L**	**min.H**
		sec.L	**sec.H**	**gio.L**	**gio.H**
		mes.L	**mes.H**	**ann.L**	**ann.H**
		gse.L	**gse.H**		
Mnemonico:	**O**	**ASCII(ore.L)**	**ASCII(ore.H)**	**ASCII(min.L)**	**ASCII(min.H)**
		ASCII(sec.L)	**ASCII(sec.H)**	**ASCII(gio.L)**	**ASCII(gio.H)**
		ASCII(mes.L)	**ASCII(mes.H)**	**ASCII(ann.L)**	**ASCII(ann.H)**
		ASCII(gse.L)	**ASCII(gse.H)**		

Vengono inizializzati i registri del Real Time Clock con ore ("ore"), minuti ("min"), secondi ("sec"), giorno ("gio"), mese ("mes"), anno ("ann") e giorno della settimana ("gse").
Tutti questi parametri devono essere trasmessi secondo una codica a nibble, in particolare prima il nibble basso (**xxx.L** = bit 0÷3) poi quello alto (**xxx.H** = bit 4÷7). I relativi valori devono essere compresi nei rispettivi range, secondo quanto indicato nella seguente tabella.

BYTE	RANGE	SIGNIFICATO
ORE	0 ... 23	ORE
MIN	0 ... 59	MINUTI
SEC	0 ... 59	SECONDI
GIO	1 ... 31	GIORNO
MES	1... 12	MESE
ANN	0 ... 99	ANNO
GSE	0 ... 6	Giorno della settimana: 0 -> DOMENICA ..... 6 -> SABATO

FIGURA 47: RANGE DI VALIDITÀ DEI BYTES DI INIZIALIZZAZIONE RTC

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:
Se si desidera inizializzare l'orologio con la data **lunedì 11 maggio 98** e l'ora **12:30:40** (corrispondente ai codici: 12, 30, 40, 11, 5, 98, 1), sarà necessario inviare la seguente sequnza: **79 12 0 14 1 8 2 11 0 5 0 2 6 1 0**.

LETTURA OROLOGIO

Sequenza del comando:

Codice Dec:	**111**
Codice Hex:	**6F**
Mnemonico:	**o**

Vengono letti e restituiti i valori dei registri del Real Time Clock.

Codici di risposta:
Vengono restituiti 7 bytes corrispondenti a ore, minuti, secondi, giorno mese, anno e giorno della settimana, nel formato illustrato nel comando precedente.
Questi vengono ricevuti secondo una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**xxx.L** = bit 0÷3) poi quello alto (**xxx.H** = bit 4÷7).

SCRITTURA NELLA SRAM SERIALE

Sequenza del comando:

Codice Dec:	**71**	**ind.L**	**ind.H**	**dat.L**	**dat.H**
Codice Hex:	**47**	**ind.L**	**ind.H**	**dat.L**	**dat.H**
Mnemonico:	**G**	**ASCII(ind.L)**	**ASCII(ind.H)**	**ASCII(dat.L)**	**ASCII(dat.H)**

Il dato "dat" (compreso nel range **0÷255**) viene memorizzato nella SRAM del modulo seriale all'indirizzo "ind" (compreso nel range **32÷255**).
Questi parametri devono essere trasmessi secondo una codica a nibble, in particolare prima il nibble basso (**xxx.L** = bit 0÷3) poi quello alto (**xxx.H** = bit 4÷7).
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:
Se si desidera scrivere, il dato 65 all' indirizzo 100 é necessario inviare la sequenza: **71 4 6 1 4**.

LETTURA DALLA SRAM SERIALE

Sequenza del comando:

Codice Dec:	**103**	**ind.L**	**ind.H**
Codice Hex:	**67**	**ind.L**	**ind.H**
Mnemonico:	**g**	**ASCII(ind.L)**	**ASCII(ind.H)**

Viene leto e restituito il byte memorizzato nella SRAM del modulo seriale.
Il valore di "ind" indica l'indirizzo di lettura; questo parametri essere compreso nel range **32÷255** e deve essere trasmessio secondo una codica a nibble, in particolare prima il nibble basso (**ind.L** = bit 0÷3) poi quello alto (**ind.H** = bit 4÷7).
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Codici di risposta:
Viene restituito il byte, compreso nel range 0÷255, tramite una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**dat.L** = bit 0÷3) poi quello alto (**dat.H** = bit 4÷7).

LINEE BUS PER LA COMUNICAZIONE 1-WIRE®

Le linee di I/O TTL disponibili sulle schede **GPC® R/T 94**, sono utilizzate per l'implementazione di due BUS di comunicazione con il protocollo **1-Wire®**. In questo modo, sfruttando i comandi illustati nel seguente paragrafo é possibile gestire i vari dispositivi sviluppati per questo tipo di standard (sensori di temperatura, memorie, Dallas iButton™, ecc.)
Vengono utilizzate le due linee di I/O TTL disponibili sul connettore CN6, con la seguente corrispondenza:

Pin 1 CN6	->	*Linea BUS 1-Wire® n.* ***0***
Pin 2 CN6	->	*Linea BUS 1-Wire® n.* ***1***

Come si potrà notare dal paragrafo seguente, i comandi ad alto livello disponibili, non supportano la presenza di più di un dispositivo per ogni linea BUS, infatti ad esempio non é presente il comando "Search ROM" relativo all'individuazione dei codici ROM dei dispositivi presenti sulla linea.
Può comunque essere gestito un collegamento di varie unità 1-Wire® in rete, sfruttando i comandi a basso livello (reset BUS, scrittura e lettura bit, scrittura e lettura bytes); in questo caso però l'implementazione risulta molto articolata e richiede anche una comunicazione pesante fra l'unità master e la **GPC® R/T94**.
E' quindi consigliato collegare alla scheda al massimo due dispositivi 1-Wire®; la figura seguente riporta un esempio di connessione con due sensori di temperatura Dallas DS18s20.

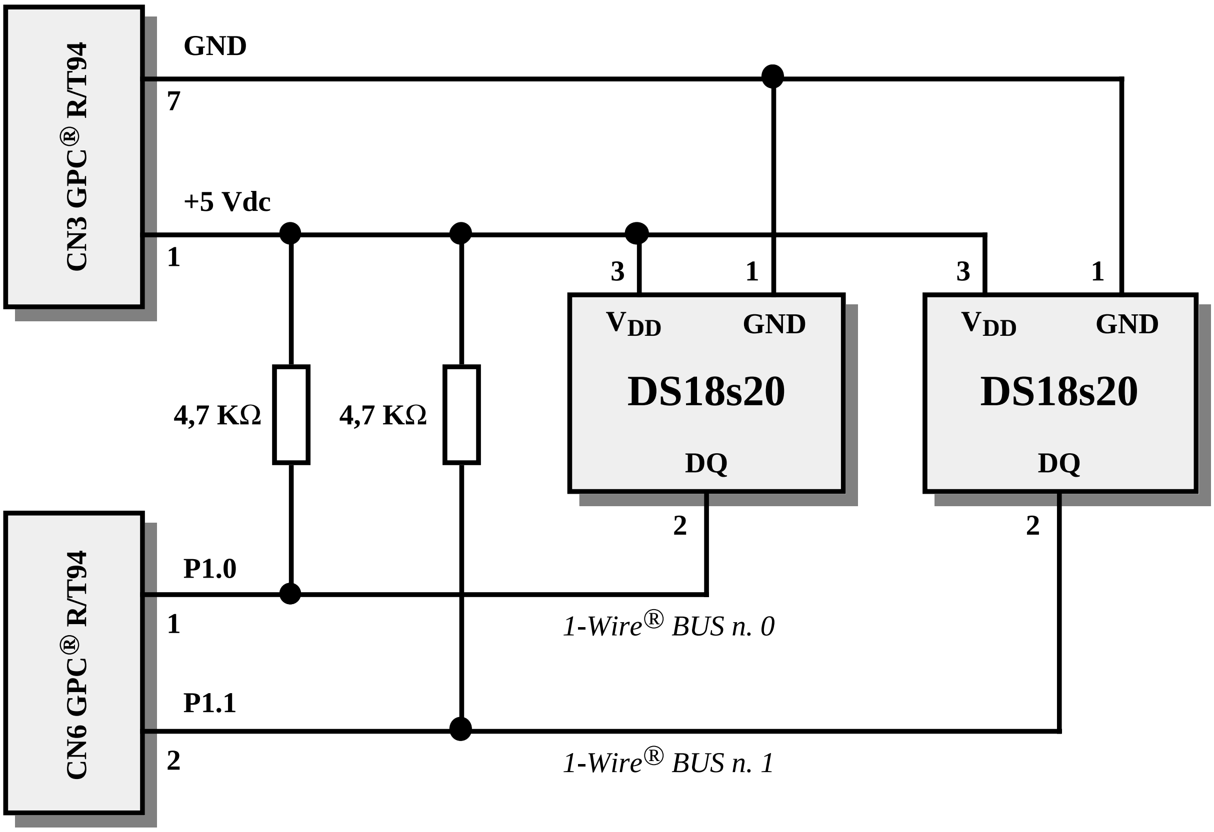


FIGURA 48: ESEMPIO DI COLLEGAMENTO CON DUE DISPOSITIVI 1-WIRE®

N.B.
Per un corretto utlizzo delle due linee BUS 1-Wire®, i jumper J5 e J6 della **GPC® R/T94**, devono essere in posizione **1-2 e 3-4**, in modo da configurare i pin di CN6 come linee di I/O generiche.

COMANDI PER LA GESTIONE DELLE LINEE BUS 1-WIRE®

Sono riportati di seguito i comandi relativi alla gestione delle due linee BUS 1-Wire®, disponibili sulle schede **GPC® R/T94**.

RESET BUS 1-WIRE® E LETTURA IMPULSO DI PRESENZA

Sequenza del comando:

Codice Dec:	**33**	**85**	**wire**
Codice Hex:	**21**	**55**	**wire**
Mnemonico:	**!**	**U**	**ASCII(wire)**

Viene inviata la sequenza di reset sulla linea BUS 1-Wire® indicata dal parametro “wire”.
Il relatio valore deve essere **0** od **1**, in caso contrario il comando viene ignorato.
Al termine della sequenza di reset viene acquisito l’implulso di presenza dall’eventuale dispositivo presente sulla linea.

Codici di risposta:
Viene restituito lo stato logico dell’impulso di presenza, in particolare:

0	*Dispositivo 1-Wire® presente e pronto a ricevere il comando*
1	*Dispositivo 1-Wire® non presente*

Esempio:
Se si desidera inviare la sequenza di reset ed acquisire l’impulso di presenza sul BUS 1-Wire® n.1, sarà necessario inviare la seguente sequenza: **33 85 1**.

SCRITTURA DI UN BIT SUL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**119**	**wire**	**bit**
Codice Hex:	**21**	**77**	**wire**	**bit**
Mnemonico:	**!**	**w**	**ASCII(wire)**	**ASCII(bit)**

Viene inviato il “bit” sulla linea BUS 1-Wire® indicata dal parametro “wire”.
Il i relativi valori devono essere **0** od **1**, in caso contrario il comando viene ignorato.

Esempio:
Se si desidera scrivere il bit 1 sulla linea BUS 1-Wire® n.0, sarà necessario inviare la seguente sequenza: **33 119 0 1**.

LETTURA DI UN BIT DAL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**114**	**wire**
Codice Hex:	**21**	**72**	**wire**
Mnemonico:	**!**	**r**	**ASCII(wire)**

Viene acquisito un bit dalla linea BUS 1-Wire® indicata dal parametro “wire”.
Il relativo valore deve essere **0** od **1**, in caso contrario il comando viene ignorato.

Codici di risposta:
Viene restituito lo stato logico (**0** o **1**) del bit acquisito dalla linea BUS indicata.

Esempio:
Se si desidera leggere un bit dalla linea BUS 1-Wire® n.0, sarà necessario inviare la seguente sequenza: **33 114 0**.

SCRITTURA DI BYTES SUL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**87**	**wire**	**n**	
		dat0.L	**dat0.H ...**	**datn.L**	**datn.H**
Codice Hex:	**21**	**57**	**wire**	**n**	
		dat0.L	**dat0.H ...**	**datn.L**	**datn.H**
Mnemonico:	**!**	**W**	**ASCII(wire)**	**ASCII(n)**	
		ASCII(dat0.L)	**ASCII(dat0.H)...**	**ASCII(datn.L)**	**ASCII(datn.H)**

Vengono inviati gli “n” bytes sulla linea BUS 1-Wire® indicata dal parametro “wire”.
Il valore di quest’ultimo deve essere **0** od **1**; il numero massimo di bytes da inviare a seguito del comando é **10**, pertanto il parametro “n” deve essere compreso nel range **1÷10**.
Gli n bytes possono essere compresi in tutto il range **0÷255** e devono essere inviati a nibble, in particolare prima il nibble basso (**datx.L** = bit 0÷3) poi quello alto (**datx.H** = bit 4÷7).
Tale comando viene ignorato nel caso in cui la sequenza contenga dei dati non validi.

Esempio:
Se si desidera scrivere i bytes 23, 118, 80 e 12 sulla linea BUS 1-Wire® n. 1, sarà necessario inviare la seguente sequenza: **33 87 1 4 7 1 6 7 0 5 12 0**.

LETTURA DI BYTES DAL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**82**	**wire**	**n**
Codice Hex:	**21**	**52**	**wire**	**n**
Mnemonico:	**!**	**R**	**ASCII(wire)**	**ASCII(n)**

Vengono letti “n” bytes dalla linea BUS 1-Wire® indicata dal parametro “wire”.

Il valore di quest'ultimo deve essere **0** od **1**; il numero massimo di bytes leggibili con un unico comando é **10**, pertanto il parametro "n" deve essere compreso nel range **1÷10**.
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Codici di risposta:
Gli n bytes, letti dalla linea BUS 1-Wire® indicata, vengono restituiti con una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**datx.L** = bit 0÷3) poi quello alto (**datx.H** = bit 4÷7).

Esempio:
Se si desidera leggere 4 bytes (che saranno 23, 118, 80 e 12) dalla linea BUS 1-Wire® n. 0, sarà necessario inviare la seguente sequenza: **33 82 0 4**. La risposta a seguito di tale comando sarà: **7 1 6 7 0 5 12 0**.

INVIO DEL COMANDO READ ROM SUL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**76**	**wire**
Codice Hex:	**21**	**4C**	**wire**
Mnemonico:	**!**	**L**	**ASCII(wire)**

A seguito di questo comando, sulla linea BUS 1-Wire® indicata dal parametro "wire", vengono effettuate le seguenti operazioni:

a) Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.

b) Se il dispositivo é presente, viene inviato il comando **Read ROM** (codice **33Hex** del protocollo 1-Wire®) ed acquisito il relativo codice ROM.

Il valore del parametro "wire" deve essere **0** od **1**, in caso contrario il comando viene ignorato.

Codici di risposta:
A seguito di questo comando vengono restituiti 8 bytes (**rom0÷rom7**) con una codifica a nibble; in particolare sarà ricevuto prima il nibble basso (**romx.L** = bit 0÷3) poi quello alto (**romx.H** = bit 4÷7).
Il significato di tali bytes é di seguito illustrato.

Dispositivo 1-Wire® presente e comando eseguito correttamente:
Gli 8 bytes corrispondono al codice ROM del dispositivo: **rom0**=Family code, **rom1÷rom6**=Serial number e **rom7**=CRC.

Dispositivo 1-Wire® non presente e comando non inviato:
Gli 8 bytes contengono tutti il valore **0**.

Esempio:
Se si vuole acquisire il codice ROM del dispositivo collegato alla linea BUS 1-Wire® n. 0, sarà necessario inviare la seguente sequenza: **33 76 0**. Ipotizzando che il codice ROM sia: 16 (family code), 56, 198, 13, 0, 8, 0, 226 (CRC), la risposta a seguito di tale comando sarà: **0 1 8 3 6 12 13 0 0 0 8 0 0 0 2 14**.

INVIO DEL COMANDO MATCH ROM SUL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**77**	**wire**			
		rom0.L	**rom0.H**	**...**	**rom7.L**	**rom7.H**
Codice Hex:	**21**	**4D**	**wire**			
		rom0.L	**rom0.H**	**...**	**rom7.L**	**rom7.H**
Mnemonico:	**!**	**M**	**ASCII(wire)**			
		ASCII(rom0.L)	**ASCII(rom0.H)...ASCII(rom7.L)**			**ASCII(rom7.H)**

A seguito di questo comando, sulla linea BUS 1-Wire® indicata dal parametro “wire”, vengono effettuate le seguenti operazioni:

a) Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.

b) Se il dispositivo é presente, viene inviato il comando **Match ROM** (codice **55Hex** del protocollo 1-Wire®) seguito dal codice ROM contenuto nei bytes: **rom0**=Family code, **rom1÷rom6**=Serial number, **rom7**=CRC.

Il valore del parametro “wire” deve essere **0** od **1**, mentre gli 8 bytes “rom0”÷“rom7”, che possono essere compresi in tutto il range **0÷255**, devono essere inviati a nibble, in particolare prima il nibble basso (**romx.L** = bit 0÷3) poi quello alto (**romx.H** = bit 4÷7).
Tale comando viene ignorato nel caso in cui la sequenza contenga dei dati non validi.

Codici di risposta:
Viene restituito un dato che può assumere i seguenti valori:

0	*Dispositivo 1-Wire® presente e comando inviato*
1	*Dispositivo 1-Wire® non presente e comando non inviato*

Esempio:
Se si vuole injviare il comando Match ROM al dispositivo collegato alla linea BUS 1-Wire® n. 1 ed ipotizzando che il codice ROM sia: 16 (family code), 56, 198, 13, 0, 8, 0, 226 (CRC), sarà necessario inviare la seguente sequenza: **33 77 1 0 1 8 3 6 12 13 0 0 0 8 0 0 0 2 14**.

INVIO DEL COMANDO SKIP ROM SUL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**83**	**wire**
Codice Hex:	**21**	**53**	**wire**
Mnemonico:	**!**	**S**	**ASCII(wire)**

A seguito di questo comando, sulla linea BUS 1-Wire® indicata dal parametro “wire”, vengono effettuate le seguenti operazioni:

a) Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.

b) Se il dispositivo é presente, viene inviato il comando **Skip ROM** (codice**CCHex** del protocollo 1-Wire®).

Il valore del parametro “wire” deve essere **0** od **1**, in caso contrario il comando viene ignorato.

Codici di risposta:
Viene restituito un dato che può assumere i seguenti valori:

0	*Dispositivo 1-Wire® presente e comando inviato*
1	*Dispositivo 1-Wire® non presente e comando non inviato*

Esempio:
Se si vuole inviare il comando Skip ROM al dispositivo collegato alla linea BUS 1-Wire® n. 1, sarà necessario inviare la seguente sequenza: **33 83 1**.

INVIO DEL COMANDO ALARM SEARCH SUL BUS 1-WIRE®

Sequenza del comando:

Codice Dec:	**33**	**65**	**wire**
Codice Hex:	**21**	**41**	**wire**
Mnemonico:	**!**	**A**	**ASCII(wire)**

A seguito di questo comando, sulla linea BUS 1-Wire® indicata dal parametro “wire”, vengono effettuate le seguenti operazioni:

a) Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.
b) Se il dispositivo é presente, viene inviato il comando **Alarm search** (codice**ECHex** del protocollo 1-Wire®) e viene determinato se questo ha il flag di allarme settato o meno.

l valore del parametro “wire” deve essere **0** od **1**, in caso contrario il comando viene ignorato.

Codici di risposta:
Viene restituito un dato che può assumere i seguenti valori:

0	*Dispositivo 1-Wire® presente con flag di allarme non settato*
1	*Dispositivo 1-Wire® presente con flag di allarme settato*
7	*Dispositivo 1-Wire® non presente e comando non inviato*

Esempio:
Se si vuole inviare il comando Alarm search al dispositivo collegato alla linea BUS 1-Wire® n. 0, sarà necessario inviare la seguente sequenza: **33 65 0**.

MODALITA’ DI COMUNICAZIONE MASTER-SLAVE A 9 BIT

La modalità Master-Slave, sfrutta la tecnica di comunicazione a 9 bit.
In particolare oltre agli 8 bit di dati, viene gestito un nono bit che serve a distinguere una chiamata, da parte della apparecchiatura "**Master**" ad una delle strutture "**Slave**", da un normale passaggio di informazioni tra il master e il dispositivo attualmente selezionato.
Quando il nono bit é posto a 1, il byte di dati deve contenere il nome, o codice di identificazione, del dispositivo con il quale si vuole comunicare, mentre ponendo questo particolare bit a 0 é poi possibile prelevare o fornire informazioni a tale dispositivo.

Nel caso particolare della comunicazione con il protocollo **ALB**, il codice di identificazione deve essere quello settato in **SETUP mode** (**NOME**).
Quando viene inviato questo byte (con il nono bit posto a 1), il dispositivo si riconosce e si pone in attesa della stringa contenente, dati o comandi (con il nono bit posto a 0); questa deve essere al massimo di **24 byte**. In questa stringa, può esserci solo un comando che comporta la restituzione di una informazione in seriale da parte del dispositivo, se ve ne é un numero superiore, i restanti comandi di questo tipo verranno ignorati.
Tra la trasmissione di un carattere ed il successivo, deve passare un tempo inferiore al tempo di **Time-Out**, in quanto, trascorso questo ritardo, si considera finita la stringa di dati ed inizia la fase di risposta. I tempi di Time-Out relativi ai vari Baud Rate sono indicati di seguito:

Baud Rate	***Time-Out***
38400 Baud	*550 µsec*
19200 Baud	*990 µsec*
9600 Baud	*1.54 msec*
4800 Baud	*3.08 msec*
2400 Baud	*6.105 msec*
1200 Baud	*12,1 msec*

Quando scade il tempo di Time-Out, inizia la sequenza di risposta; questa constiste in un byte contenente il codice di presenza **6** (**6 Hex**), oppure da un dato o sequenza di dati, relativi ad un comando di lettura inviato nella chiamata precedente.

Per esempio, se viene trasmessa una stringa contenente il comando di lettura Port, si avrà che in quella particolare chiamata sarà restituito il codice di presenza, mentre nella successiva, verrà trasmesso il dato acquisito dal Port, richiesto in precedenza.

Il master dopo aver aver completato la trasmissione dell’ultimo carattere della stringa, dovrà attendere un tempo di:

"Tempo di trasmissione di un carattere" + Time-Out

prima che arrivi il primo carattere della stringa di risposta, trasmessa dal dispositivo selezionato.

Per esempio, se si lavora a 38.4 KBaud, quando é stata completata la tramissione dell’ultimo carattere, é necessario attendere un tempo di circa 810 µsec, prima che sia completata la ricezione del primo dato di risposta trasmesso dal dispositivo.

NOTE:

1) Tra una chiamata e la successiva, per avere la certezza che il comando trasmesso sia correttamente eseguito, é neccessario attendere un tempo che é funzione del numero di comandi inviati e del tipo di operazioni che questi comportano.

2) Se l'unità master di controllo non é in grado di dialogare a 9 bit, é possibile simulare questo tipo di comunicazione, sfruttando il bit di parità e programmando, prima di trasmettre ogni singolo byte, la parità pari o dispari, secondo quanto indicato di seguito:

Il Byte da trasmettere ha un numero PARI di bit a 1

Se il Bit 9 deve essere ***1***	->	*Programmare la parità DISPARI*
Se il Bit 9 deve essere ***0***	->	*Programmare la parità PARI*

Il Byte da trasmettere ha un numero DISPARI di bit a 1

Se il Bit 9 deve essere ***1***	->	*Programmare la parità PARI*
Se il Bit 9 deve essere ***0***	->	*Programmare la parità DISPARI*

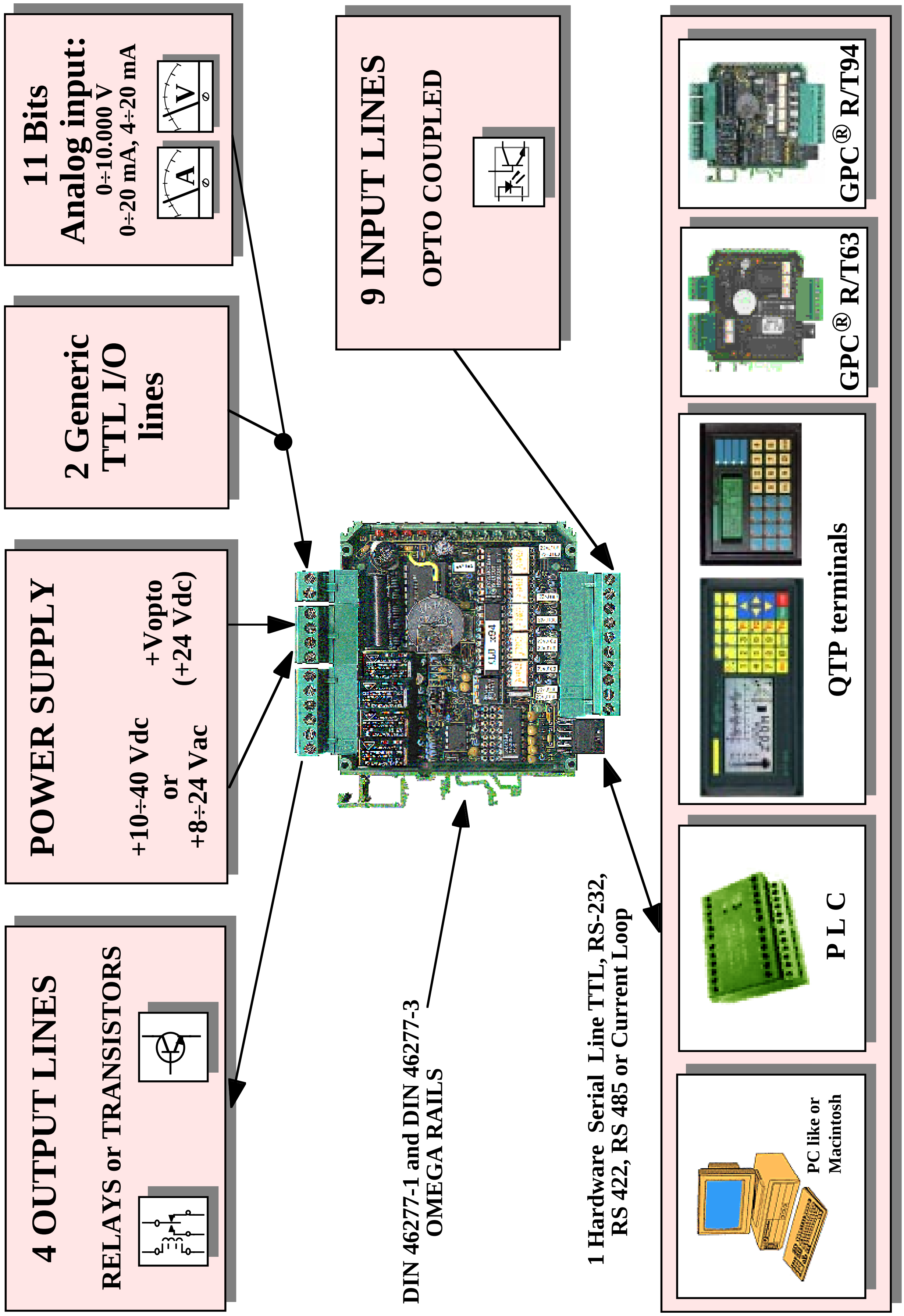


FIGURA 49: SCHEMA DELLE POSSIBILI CONNESSIONI

BIBLIOGRAFIA

E' riportato di seguito, un elenco di manuali e note tecniche, a cui l'utente può fare riferimento per avere maggiori chiarimenti, sui vari componenti montati a bordo della scheda **GPC® R/T94**.

Manuale TEXAS INSTRUMENTS: *The TTL Data Book - SN54/74 Families*
Manuale TEXAS INSTRUMENTS: *RS-422 and RS-485 Interface Circuits*

Manuale MAXIM: *New Releases Data Book - Volume 4*

Manuale XICOR: *Data Book*

Manuale PHILIPS: *IC12 - I²C bus*

Manuale ATMEL: *Microcontroller - AT89 series*

Manuale TOSHIBA: *Photo Couplers - Data Book*

Manuale MOTOROLA: *Bipolar Power Transistor Data*

Manuale HEWLETT PACKARD: *Optoelectronics Designer's Catalog*

Note Tecniche NATIONAL: *LM2825N*

Note Tecniche DALLAS: *Documentazione ed applications notes relative ai dispositivi ed al protocollo di comunicazione 1-Wire®*

Per avere tutti gli aggiornamenti di tali manuali e di tutti i data-sheets fare riferimento anche ai siti INTERNET delle case madri costruttrici.

APPENDICE A: COMANDI DEL FIRMWARE ALB X94

Sono riportate di seguito le tabelle riassuntive con i codici dei comandi del firmware **ALB x94.**

COMANDO	Cod. DEC.	Cod. HEX	MNEMONICO
Reset Generale	65 97	41 61	A a
Rischiesta EEPROM disponibile	66	42	B
Scrittura del Byte di Presenza	89 dat.L dat.H	59 dat.L dat.H	Y ASCII(dat.L) ASCII(dat.H)
Lettura del Byte di Presenza	121	79	y
Lettura versione	86	56	V
Lettura cod. scheda	75	4B	K
Settaggio config. del pin AUX	85 cfg	55 cfg	U ASCII(cfg)
Lettura config. del pin AUX	102	66	f
Scrittura del Port di OUTPUT	87 1 dat 0	57 1 dat 0	W SOH ASCII(dat) NUL
Lettura del Port di INPUT	82 0	52 0	R NUL
SET Port.Bit	83 1 bit	53 1 bit	S SOH ASCII(bit)
SET Port.Bit temporizzato	115 1 bit tmp.L tmp.H	73 1 bit tmp.L tmp.H	s SOH ASCII(bit) ASCII(tmp.L) ASCII(tmp.H)
CLEAR Port.Bit	67 1 bit	43 1 bit	C SOH ASCII(bit)
CLEAR Port.Bit temporizzato	99 1 bit tmp.L tmp.H	63 1 bit tmp.L tmp.H	c SOH ASCII(bit) ASCII(tmp.L) ASCII(tmp.H)
Lettura Port.Bit	114 port bit	72 port bit	r ASCII(port) ASCII(bit)
Lettura Port.Bit con deboucing	68 port bit tmp.L tmp.H	44 port bit tmp.L tmp.H	D ASCII(port) ASCII(bit) ASCII(tmp.L) ASCII(tmp.H)
Onda Quadra da "1" su Port.Bit temporizzata	112 1 bit tmp.L tmp.H sta.L sta.H	70 1 bit tmp.L tmp.H sta.L sta.H	p SOH ASCII(bit) ASCII(tmp.L) ASCII(tmp.H) ASCII(sta.L) ASCII(sta.H)
Onda Quadra da "0" su Port.Bit temporizzata	119 1 bit tmp.L tmp.H sta.L sta.H	77 1 bit tmp.L tmp.H sta.L sta.H	w SOH ASCII(bit) ASCII(tmp.L) ASCII(tmp.H) ASCII(sta.L) ASCII(sta.H)
Onda Quadra su Port.Bit	80 1 bit tmp.L tmp.H	50 1 bit tmp.L tmp.H	P SOH ASCII(bit) ASCII(tmp.L) ASCII(tmp.H)

FIGURA A1: TABELLA 1 DEI COMANDI ALB X94

COMANDO	Cod. DEC.	Cod. HEX	MNEMONICO
Lettura Counter	73	49	I
Reset Counter	88 120	58 78	X x
Numero ultimo mess.	77	4D	M
Memorizzazione Messaggio	69 msg.L msg.H car0.L car0.H car9.L car9.H	45 msg.L msg.H car0.L car0.H car9.L car9.H	E ASCII(msg.L) ASCII(msg.H) ASCII(car0.L) ASCII(car0.H) .... ASCII(car9.L) ASCII(car9.H)
Lettura Messaggio	76 msg.L msg.H	4C msg.L msg.H	L ASCII(msg.L) ASCII(msg.H)
Settaggio RTC	79 ore.L ore.H min.L min.H sec.L sec.H gio.L gio.H mes.L mes.H ann.L ann.H gse.L gse.H	4F ore.L ore.H min.L min.H sec.L sec.H gio.L gio.H mes.L mes.H ann.L ann.H gse.L gse.H	O ASCII(ore.L) ASCII(ore.H) ASCII(min.L) ASCII(min.H) ASCII(sec.L) ASCII(sec.H) ASCII(gio.L) ASCII(gio.H) ASCII(mes.L) ASCII(mes.H) ASCII(ann.L) ASCII(ann.H) ASCII(gse.L) ASCII(gse.H)
Lettura RTC	111	6F	o
Scrittura SRAM RTC	71 ind.L ind.H dat.L dat.H	47 ind.L ind.H dat.L dat.H	G ASCII(ind.L) ASCII(ind.H) ASCII(dat.L) ASCII(dat.H)
Lettura SRAM RTC	103 ind.L ind.H	67 ind.L ind.H	g ASCII(ind.L) ASCII(ind.H)
Reset BUS 1-Wire®	33 85 wire	21 55 wire	! U ASCII(wire)
WR bit BUS 1-Wire®	33 119 wire bit	21 77 wire bit	! w ASCII(wire) ASCII(bit)
RD bit BUS 1-Wire®	33 114 wire	33 72 wire	! r ASCII(wire)
Scrittura n bytes sul BUS 1-Wire®	33 87 wire n dat0.L dat0.H .... datn.L datn.H	21 57 wire n dat0.L dat0.H .... datn.L datn.H	! W ASCII(wire) ASCII(n) ASCII(dat0.L) dat(dat0.H) .... ASCII(datn.L) ASCII(datn.H)
Lettura n bytes dal BUS 1-Wire®	33 82 wire n	21 52 wire n	! R ASCII(wire) ASCII(n)
Read ROM (1-Wire®)	33 76 wire	33 4C wire	! L ASCII(wire)
Match ROM (1-Wire®)	33 77 wire rom0.L rom0.H rom7.L rom7.H	33 4D wire rom0.L rom0.H rom7.L rom7.H	! M ASCII(wire) ASCII(rom0.L) ASCII(rom0.H) .... ASCII(rom7.L) ASCII(rom7.H)
Skip ROM (1-Wire®)	33 83 wire	33 53 wire	! S ASCII(wire)
Alarm search (1-Wire®)	33 65 wire	33 41 wire	! A ASCII(wire)

FIGURA A2: TABELLA 2 DEI COMANDI ALB x94

Features

- Compatible with MCS-51™ Products
- 4K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 3.0V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Brown-Out Detection

8-Bit Microcontroller with 4K Bytes Flash

AT89C4051 Preliminary

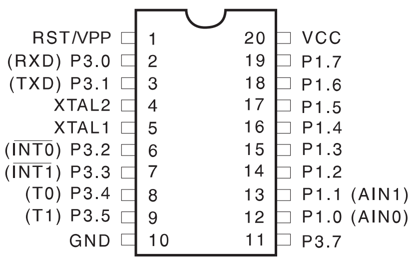
Description

The AT89C4051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 4K Bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C4051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C4051 provides the following standard features: 4K Bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C4051 is designed with static logic for operation down to zero frequency and supports two software-selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

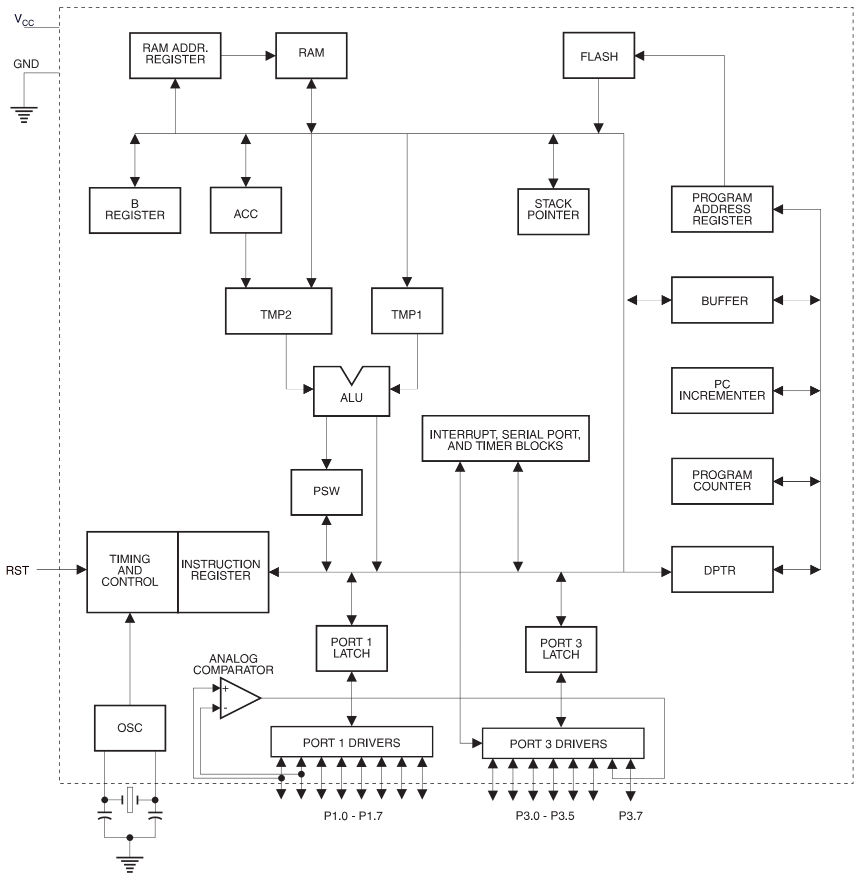
Pin Configuration

PDIP/SOIC



Rev. 1001A–02/98

Block Diagram



Pin Description

V_{CC}
Supply voltage.

GND
Ground.

Port 1
Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3
Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C4051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and verification.

RST
Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

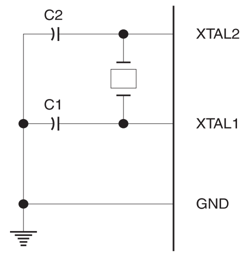
XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier.

Oscillator Characteristics

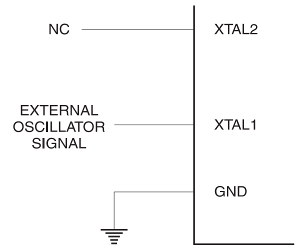
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C4051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Restrictions on Certain Instructions

The AT89C4051 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 4K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 4K for the AT89C4051. This should be the responsibility of the software programmer. For example, LJMP 0FE0H would be a valid instruction for the AT89C4051 (with 4K of memory), whereas LJMP 1000H would not.

1. Branching instructions:
LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to FFFH for the 89C4051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:
The AT89C4051 contains 128 bytes of internal data memory. Thus, in the AT89C4051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes(1)

Program Lock Bits	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

PROGRAMMER'S GUIDE AND INSTRUCTION SET

Memory Organization

Program Memory
The 80C51 has separate address spaces for program and data memory. The Program memory can be up to 64k bytes long. The lower 4k can reside on-chip. Figure 1 shows a map of the 80C51 program memory.

The 80C51 can address up to 64k bytes of data memory to the chip. The MOVX instruction is used to access the external data memory.

The 80C51 has 128 bytes of on-chip RAM, plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 2 shows the Data Memory organization.

Direct and Indirect Address Area
The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into three segments as listed below and shown in Figure 3.

1. Register Banks 0-3: Locations 0 through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers 0 through 7. Reset initializes the stack pointer to location 07H, and it is incremented once to start from location 08H, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM where it is not used for data storage (i.e., the higher part of the RAM).
2. Bit Addressable Area: 16 bytes have been assigned for this segment, 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH). The bits can be referred to in two ways, both of which are acceptable by most assemblers. One way is to refer to their address (i.e., 0-7FH). The other way is with reference to bytes 20H to 2FH. Thus, bits 0-7 can also be referred to as bits 20.0-20.7, and bits 8-FH are the same as 21.0-21.7, and so on. Each of the 16 bytes in this segment can also be addressed as a byte.
3. Scratch Pad Area: 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

Figure 2 shows the different segments of the on-chip RAM.

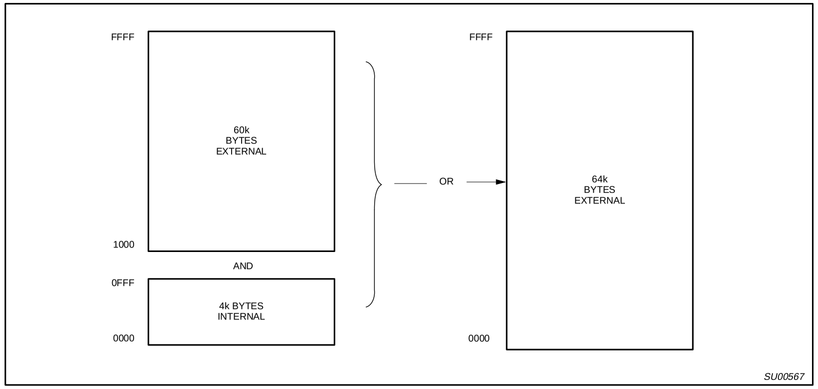


Figure 1. 80C51 Program Memory

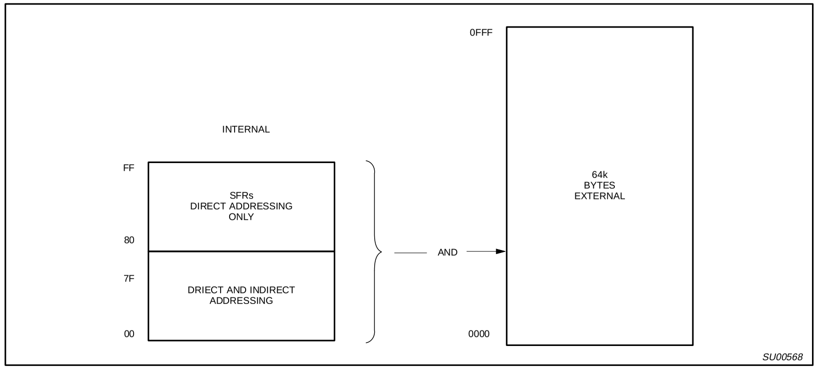


Figure 2. 80C51 Data Memory

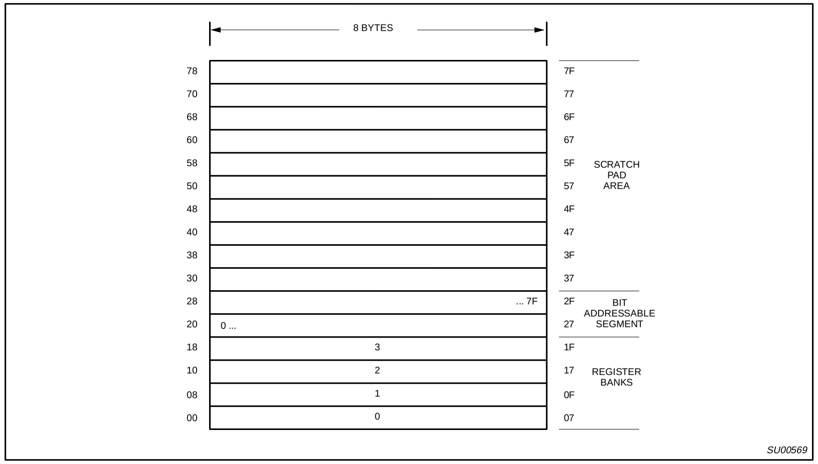


Figure 3. 128 Bytes of RAM Direct and Indirect Addressable

Philips Semiconductors

80C51 Family

80C51 family programmer's guide and instruction set

Table 1. 80C51 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	–	–	ES	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt priority	B8H	–	–	–	PS	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	–	–	–	–	–	–	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	Rxd	FFH
PCON[1]	Power control	87H	SMOD	–	–	–	GF1	GF0	PD	IDL	0xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	–	P	00H
SBUF	Serial data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TMOD	Timer mode	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H

NOTES:
* Bit addressable
1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented on the NMOS 8051/8031.

Philips Semiconductors

80C51 Family

80C51 family programmer's guide and instruction set

8 BYTES

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

BIT ADDRESSABLE

SU00570

Figure 4. SFR Memory Map

Those SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture Chapter of this book.

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	–	P

CY PSW.7 Carry Flag.
AC PSW.6 Auxiliary Carry Flag.
F0 PSW.5 Flag 0 available to the user for general purpose.
RS1 PSW.4 Register Bank selector bit 1 (SEE NOTE 1).
RS0 PSW.3 Register Bank selector bit 0 (SEE NOTE 1).
OV PSW.2 Overflow Flag.
– PSW.1 Usable as a general purpose flag.
P PSW.0 Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bus in the accumulator.

NOTE:
1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	–	–	–	GF1	GF0	PD	IDL

SMOD Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.
– Not implemented, reserved for future use.*
– Not implemented reserved for future use.*
– Not implemented reserved for future use.*
GF1 General purpose flag bit.
GF0 General purpose flag bit.
PD Power Down Bit. Setting this bit activates Power Down operation in the 80C51. (Available only in CMOS.)
IDL Idle mode bit. Setting this bit activates Idle Mode operation in the 80C51. (Available only in CMOS.)

If 1s are written to PD and IDL at the same time, PD takes precedence.

* User software should not write 1s to reserved bits. These bits may be used in future 8051 products to invoke new features.

INTERRUPTS:

To use any of the interrupts in the 80C51 Family, the following three steps must be taken.

1. **Set the EA (enable all) bit in the IE register to 1.**
2. **Set the corresponding individual interrupt enable bit in the IE register to 1.**
3. **Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.**

INTERRUPT SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H

In addition, for external interrupts, pins INT0 and INT1 (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITx = 0 level activated

ITx = 1 transition activated

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	–	–	ES	ET1	EX1	ET0	EX0

EA IE.7 Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
— IE.6 Not implemented, reserved for future use.*
— IE.5 Not implemented, reserved for future use.*
ES IE.4 Enable or disable the serial port interrupt.
ET1 IE.3 Enable or disable the Timer 1 overflow interrupt.
EX1 IE.2 Enable or disable External Interrupt 1.
ET0 IE.1 Enable or disable the Timer 0 overflow interrupt.
EX0 IE.0 Enable or disable External Interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

ASSIGNING HIGHER PRIORITY TO ONE OR MORE INTERRUPTS:

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.

Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

PRIORITY WITHIN LEVEL:

Priority within level is only to resolve simultaneous requests of the same priority level.

From high to low, interrupt sources are listed below:

IE0
TF0
IE1
TF1
RI or TI

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

–	–	–	PS	PT1	PX1	PT0	PX0

–	IP.7	Not implemented, reserved for future use.*
–	IP.6	Not implemented, reserved for future use.*
–	IP.5	Not implemented, reserved for future use.*
PS	IP.4	Defines the Serial Port interrupt priority level.
PT1	IP.3	Defines the Timer 1 interrupt priority level.
PX1	IP.2	Defines External Interrupt 1 priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level.
PX0	IP.0	Defines the External Interrupt 0 priority level.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

Philips Semiconductors

80C51 Family — 80C51 family programmer's guide and instruction set

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Timer 1				**Timer 0**			

GATE	When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).
C/$\overline{T}$	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
M1	Mode selector bit. (NOTE 1)
M0	Mode selector bit. (NOTE 1)

NOTE 1:

M1	**M0**	**Operating Mode**	
0	0	0	13-bit Timer (8048 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standart Timer 0 control bits. TH0 is an8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

TIMER SET-UP

Tables 2 through 5 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. If it is desired to run Timers 0 and 1 simultaneously, in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control), and Timer 1 in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 2 ORed with 60H from Table 5).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that at a different point in the program by setting bit TRx (in TCON) to 1.

TIMER/COUNTER 0

Table 2. As a Timer:

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	Two 8-bit Timers	03H	0BH

Table 3. As a Counter:

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	One 8-bit Counter	07H	0FH

NOTES:
1. The timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{INT0}$ (P3.2) when TR0 = 1 (hardware control).

TIMER/COUNTER 1

Table 4. As a Timer:

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	Does not run	30H	B0H

Table 5. As a Counter:

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	Not available	–	–

NOTES:
1. The timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{INT1}$ (P3.2) when TR1 = 1 (hardware control).

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier. (NOTE 1)
SM1	SCON.6	Serial Port mode specifier. (NOTE 1)
SM2	SCON.5	Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 6.)
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	$F_{OSC.}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$F_{OSC.}/64$ or $F_{OSC.}/32$
1	1	3	9-bit UART	Variable

SERIAL PORT SET-UP:

Table 6.

MODE	SCON	SM2 VARIATION
0 1 2 3	10H 50H 90H D0H	Single Processor Environment (SM2 = 0)
0 1 2 3	NA 70H B0H F0H	Multiprocessor Environment (SM2 = 1)

GENERATING BAUD RATES

Serial Port in Mode 0:

Mode 0 has a fixed baud rate which is 1/12 of the oscillator frequency. To run the serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} \quad \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1:

Mode 1 has a variable baud rate. The baud rate is generated by Timer 1.

USING TIMER/COUNTER 1 TO GENERATE BAUD RATES:

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

$$\text{Baud Rate} \quad \frac{K}{32} \quad \frac{\text{Osc Freq}}{12 \quad [256 \quad (TH1)]}$$

If SMOD = 0, then K = 1.
If SMOD = 1, then K = 2 (SMOD is in the PCON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.

$$TH1 \quad 256 \quad \frac{K}{384} \quad \frac{\text{Osc Freq}}{\text{baud rate}}$$

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (i.e., ORL PCON,#80H). The address of PCON is 87H.

SERIAL PORT IN MODE 2:

The baud rate is fixed in this mode and is 1/32 or 1/64 of the oscillator frequency, depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.

SMOD = 0, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit: ORL PCON,#80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3:

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

80C51 FAMILY INSTRUCTION SET

Table 7. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings[1]

Instruction	C	OV	AC	Instruction	C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@Ri	8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
#data	8-bit constant included in the instruction.
#data 16	16-bit constant included in the instruction
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with carry	1	12
ADDC	A,#data	Add immediate data to A_{CC} with carry	2	12
SUBB	A,Rn	Subtract Register from A_{CC} with borrow	1	12
SUBB	A,direct	Subtract direct byte from A_{CC} with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from A_{CC} with borrow	1	12
SUBB	A,#data	Subtract immediate data from A_{CC} with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS (Continued)				
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A and B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator left	1	12
RLC	A	Rotate Accumulator left through the carry	1	12
RR	A	Rotate Accumulator right	1	12
RRC	A	Rotate Accumulator right through the carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
DATA TRANSFER (Continued)				
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	RN,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A_{CC}	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to A_{CC}	1	24
MOVX	A,@Ri	Move external RAM (8-bit addr) to A_{CC}	1	24
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A_{CC}	1	24
MOVX	A,@Ri,A	Move A_{CC} to external RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move A_{CC} to external RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order digit indirect RAM with A_{CC}	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to carry	2	24
ANL	C,/bit	AND complement of direct bit to carry	2	24
ORL	C,bit	OR direct bit to carry	2	24
ORL	C,/bit	OR complement of direct bit to carry	2	24
MOV	C,bit	Move direct bit to carry	2	12
MOV	bit,C	Move carry to direct bit	2	24
JC	rel	Jump if carry is set	2	24
JNC	rel	Jump if carry not set	2	24

Philips Semiconductors

80C51 Family — 80C51 family programmer's guide and instruction set

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
BOOLEAN VARIABLE MANIPULATION (Continued)				
JB	rel	Jump if direct bit is set	3	24
JNB	rel	Jump if direct bit is not set	3	24
JBC	bit,rel	Jump if direct bit is set and clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute subroutine call	2	24
LCALL	addr16	Long subroutine call	3	24
RET		Return from subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute jump	2	24
LJMP	addr16	Long jump	3	24
SJMP	rel	Short jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is zero	2	24
JNZ	rel	Jump if Accumulator is not zero	2	24
CJNE	A,direct,rel	Compare direct byte to A_{CC} and jump if not equal	3	24
CJNE	A,#data,rel	Compare immediate to A_{CC} and jump if not equal	3	24
CJNE	RN,#data,rel	Compare immediate to register and jump if not equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	24
DJNZ	Rn,rel	Decrement register and jump if not zero	2	24
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	24
NOP		No operation	1	12

APPENDICE C: INDICE ANALITICO

E

F

I

J

L

M

N

O

P

R

S

T

U

V